Separable Inverse Problems, Blind Deconvolution, and Stray Light Correction for Extreme Ultraviolet Solar Images

by

Paul R. Shearer

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Applied and Interdisciplinary Mathematics)
in The University of Michigan
2013

Doctoral Committee:

Associate Research Scientist Richard A. Frazin, Co-Chair
Professor Anna C. Gilbert, Co-Chair
Professor Alfred O. Hero III, Co-Chair
Professor Selim Esedoglu
Professor Jeffrey A. Fessler

To my wife

ACKNOWLEDGEMENTS

The work in this thesis benefitted immeasurably from the contributions of the extraordinary people who took it upon themselves to advise me, as well as family and friends who supported me throughout the process.

My application field co-advisor, Dr. Richard Frazin of the Department of Atmospheric, Oceanic, and Space Science, is responsible for the idea of correcting stray light in the Extreme Ultraviolet Imager (EUVI) instruments using lunar transits and early mission data. Giving a Ph.D. student such an interesting, significant, and tractable problem is the best thing an advisor can do, and for this I am very grateful. He has challenged me to meet high standards of scientific rigor and clarity, but perhaps most importantly, he has patiently encouraged and helped me to meet those standards. This can be an overwhelming task for a beginning student and Dr. Frazin has helped me to break it down into smaller, more manageable steps. Writing up the stray light correction work for astrophysical journals required me to develop a simpler communication style accessible to non-mathematicians, and his advice on this has been invaluable. He has also organized meetings and brought me with him to conferences to promote my work.

My mathematics co-advisor, Professor Anna C. Gilbert, has been a great inspiration and influence throughout my graduate school years. She presents even the most challenging ideas accessibly and enthusiastically, in a way that seems to inevitably lead to an intuitive grasp and a love for any topic. It was her influence most of all that led me to do work in signal processing and optimization algorithms. She has read through endless drafts and greatly helped me to clarify and organize my thoughts. Any time a referee calls a paper of mine well-written, it is a credit to her.

Finally, my work has benefitted greatly from Professor Alfred O. Hero III's patient guidance, vast background, and writing advice. His input on the stray light correction project was very much appreciated, and his critiques improved the variable elimination paper considerably. Going to his group meetings is always an interesting experience full of new ideas. I hope to work with him more on related topics in signal

processing and machine learning in the future.

J.-P. Wuelser (Lockheed Martin Solar & Astrophysics Lab), Frederic Auchére (Institut d'Astrophysique Spatiale), and Raymond Mercier (Institut d'Optique) contributed greatly to the EUVI project through many long discussions. I thank Dr. Dan Seaton for suggesting and supporting a fruitful collaboration at the Royal Observatory of Belgium.

I appreciate the service of Professors Selim Esedoglu and Jeffrey A. Fessler on my thesis committee, and am grateful for their interest in my work. I would like to thank Paul Kessenich and Karen Rhea for helping me to develop as a teacher. I enjoyed working with Dr. Mark Conger and Shashir Reddy to teach high school students about Math and the Internet.

Many people have given me a great deal of personal support and encouragement throughout graduate school. More than anyone my wife has made the whole journey worthwhile. My parents have supported me in good times and bad. I hope that Dr. Daniel J. Kneezel, Dr. Patrick Rooney, and future Drs. Ari Shnidman and Zachary Scherr enjoyed living with me in 'the uninhabitable house' as much as I did with them. More friends than I can ever name brightened the way, and I would like to thank everyone who hosted social events and everyone who came to those at our house.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF ALGORITHMS

Algorithm

ABSTRACT

Separable inverse problems, blind deconvolution, and stray light correction for extreme ultraviolet images
by
Paul Richard Shearer

Co-Chairs: Anna C. Gilbert, Richard A. Frazin, and Alfred O. Hero III

Given a system that maps inputs to outputs, the determination of the unknown input that gave rise to a known output is called an inverse problem. Most real systems include noise and other degradations which prevent exact solution of an inverse problem, but an approximate solution can be obtained by statistical inference. This thesis considers the special class of *separable* inverse problems in which the system is linear but incompletely characterized. The focus is primarily on methods for image deblurring by blind and semiblind deconvolution, and much of the work is devoted to a particular scientific problem in solar imaging. However, many of the ideas and methods are more generally applicable.

In separable inverse problems, the output data b is modeled as a linear transformation A of the unknown input z^{true} plus noise ϵ. The linear transformation A depends upon unknown side parameters y^{true}, so that $b = A(y^{\text{true}})z^{\text{true}} + \epsilon$ where y^{true} and z^{true} are to be determined. In blind and semiblind deconvolution problems, b is a blurry image, z^{true} is the unknown sharp image, A represents convolution with the imaging system's point spread function (PSF), and y^{true} denotes the PSF or some parameters determining it. A solution is commonly obtained by optimizing a cost function $F(y, z)$ derived from maximum likelihood or Bayesian estimation theory, possibly with constraints on y and z.

The first contribution of the thesis is a generalization of the popular variable elimination technique for optimizing $F(y, z)$. When the optimization problem is of

unconstrained least squares form, the optimal value of z may be expressed in terms of y using the pseudoinverse of A, and $F(y, z)$ may be expressed in terms of y alone. Optimizing this reduced cost function over y often leads to a faster, more accurate solution than optimizing $F(y, z)$ directly, particularly when F exhibits ill-conditioning or other pathologies. However, the pseudoinverse formula for z is not applicable to problems with nonquadratic likelihoods, inequality constraints on z, or other common departures from least squares. A new class of *semi-reduced* optimization methods are proposed to overcome this limitation. These methods are obtained by modifying standard optimization methods to behave as though a variable has been eliminated. Semi-reduced methods encompass several existing variable elimination techniques, but can be used to solve problems with inequality constraints and nonquadratic likelihoods as well. New linear algebra techniques are proposed to enable semi-reduced methods to share some of the best features of variable elimination methods. Tests on difficult exponential sum fitting and blind deconvolution problems indicate that the proposed approach can have significant speed and robustness advantages over standard methods.

The second contribution is a new method for blind deconvolution of blurry images corrupted by camera shake. The method first determines the PSF, then performs nonblind deconvolution of the blurry image. The PSF is determined by exploiting the fact that strong edges are generally much sparser in a sharp image than in a blurry one. The PSF and the sharp image's edge map are determined simultaneously by an alternating projected gradient optimization, where the edge map is subjected to an initially stringent sparsity constraint that is slowly relaxed. In experiments on a standard test set, the proposed method is faster than the state-of-the-art variational Bayes methods and competitive in deblurring performance.

The third contribution is the determination of PSFs for EUVI-A and EUVI-B, the two extreme ultraviolet (EUV) solar imaging instruments aboard NASA's STEREO mission. The PSFs are determined for all four filter bands (171, 195, 284, and 304 Å) and are used to correct long-range scattering effects that contaminate the images with a haze of stray light. The PSFs are modeled using semi-empirical parametric formulas, and their parameters are determined by semiblind deconvolution of EUVI images. The EUVI-B PSFs are determined by semiblind deconvolution of lunar transit data, exploiting the fact that the Moon is not a significant EUV source. The EUVI-A PSFs are determined by analysis of simultaneous A/B observations from December 2006, when the two EUVIs had nearly identical lines of sight to the Sun. This is the first EUV stray light correction derived by applying statistical inference to a mathematical image formation model that accounts for long-range scatter and noise. This model-based approach enables the calculation of the first error estimates for deconvolved EUV images.

CHAPTER 1

Introduction

Many natural and man-made systems have mathematical models that describe their output, response, or reaction when a given input, stimulus, or change is applied. The task of calculating the output from a given input is called a *forward problem*, while determining the input from a given output is called an *inverse problem*.

Inverse problems are generally harder than forward problems for two reasons. First, few mathematical operations are invertible in closed form, and outside these cases one must resort to iterative methods with no general guarantee of success. Second, many operations cause an irreversible loss of information about the input, and one can only hope to determine a most likely or most reasonable input among many possible candidates. To do this we take the common approach of defining a cost function derived from maximum likelihood or Bayesian estimation theory, and seeking a minimizer of this function over the space of possible inputs [105].

In this thesis we present several contributions to inverse problems from the text of published or soon-to-be-published journal and conference articles. The status of these articles and their co-authors are given in a footnote at the beginning of each chapter. These papers are written as self-contained contributions for specific audiences in mathematics, image processing, and astrophysics, so each uses a different notation and style. This chapter overviews the type of problems we studied, technical challenges we encountered, solution methods we found particularly useful, and our specific contributions.

1.1 Linear inverse problems

Linear inverse problems are those in which the output is generated by a linear system. In this case, the output b obeys

$$b = Az^{\text{true}} + \epsilon, \tag{1.1}$$

where A is a linear operator, z^{true} is the unknown input, and ϵ is noise. The associated optimization typically takes the form

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad F(z) \triangleq L(Az) + R_z(z), \tag{1.2}$$

where z represents an estimate of z^{true}, $\mathcal{Z}$ is the set of admissible values of z, $L(Az)$ measures the discrepancy between Az and b, and $R_z(z)$ is a penalty function that takes higher values when z takes on more unlikely or unreasonable characteristics. The best results are often obtained when L is chosen in accordance with statistical principles. Given the distribution of ϵ, one typically sets L to be the negative log-likelihood $L(Az) = -\log p(b \,|\, Az)$, where $p(b \,|\, Az)$ is the probability of observing b given the value of Az. The function $R_z(z)$ can be interpreted as the negative logarithm of a prior distribution on z (the Bayesian *maximum a posteriori* interpretation), or simply as a penalty (the *penalized likelihood* interpretation). In many linear inverse problems the functions L and R_z are convex, which implies immediately that $F(z)$ is convex [16]. This means that all local minima are global minima, and if $F(z)$ is strictly convex the minimum is unique.

Imaging systems are a particularly rich source of linear inverse problems. A central problem in this work is image deblurring by deconvolution. In this case b is the blurry image (an array of pixel intensities), z^{true} is the ideal blur-free image, and the operator A blurs z^{true} by convolving it with the system's point spread function (PSF), the system's response to a unit intensity point source. (We assume throughout that the PSF is the same everywhere in the imaging plane.) Another important problem is tomography, the creation of a 3D object reconstruction from 2D projections. Tomography has applications ranging from medical imaging [74] to solar physics [39]. In tomography the linear operator calculates integrals of object emissions along a given line of sight.

A perennial challenge of imaging problems is their high dimensionality. Even a modest 256 × 256 unknown image z contains 65,536 free variables, and most images are much larger. Optimization at this scale requires special techniques: even optimization of a positive definite quadratic function is nontrivial, as it requires the solution of a large linear system. Direct linear algebra methods such as Gaussian elimination become impractical due to memory requirements, so iterative methods such as conjugate gradients (CG) must be used. These methods can be very slow when A is ill-conditioned and can suffer from accumulated roundoff error [101]. Limited memory optimization methods, such as L-BFGS, truncated Newton, and nonlinear conjugate gradients, either rely on CG or are closely related to it and thus suffer from similar limitations [82]. Issues of conditioning may sometimes be alleviated by a good preconditioner that exploits the structure of A, or by a prudent choice of the regularizing penalty.

A more recent challenge in imaging inverse problems the treatment of hard constraints and nonsmooth penalties. Simply enforcing a nonnegativity constraint on z can substantially improve performance [8], and nonsmooth penalties such as the ℓ_1 or total variation norms are central tools in advanced image processing [36]. Many classical optimization methods do not adapt well to this case, but projected gradient and proximal splitting methods, which take advantage of the structure of (1.2), are among the most successful for such problems [28].

1.2 Separable inverse problems and blind deconvolution

In this work we are concerned with *separable* inverse problems, a generalization of linear inverse problems where the linear system itself contains unknowns [45]. Separable problems have the form

$$b = A(y^{\text{true}})z^{\text{true}} + \epsilon, \tag{1.3}$$

where, in contrast to the linear case, the system matrix A now depends on unknown parameters y^{true}. These new unknowns create difficulties not seen in linear inverse problems. The cost function becomes

$$\underset{y\in\mathcal{Y},z\in\mathcal{Z}}{\text{minimize}} \quad F(y,z) \triangleq L(A(y)z) + R_y(y) + R_z(z), \tag{1.4}$$

and is generally no longer convex, introducing the possibility of an iterative method getting stuck in a poor local minimum. The dependence on y can also introduce severe ill-conditioning and pathologies that slow down the discovery of even a local minimum. In some problems $A(y)$ is a linear function of y, but in others the dependence is highly nonlinear. Various methods for solving this problem are discussed in Chapter 2.

Despite their difficulty, separable inverse problems receive much attention due to the breadth and impact of their applications. Spectroscopy, a fundamental measurement technique in chemistry and biology, involves separable problems during data analysis [79]. When b and z are matrices and $A(y) = y$, we recover the problem of matrix factorization, an intensely-researched problem in signal processing and machine learning [27]. Other applications include motion-blur compensation in PET imaging [57], Stokes imaging with phase diversity [103], and functional MRI [83]. Many other applications are discussed in [45].

Much of this thesis concerns the problems of *blind* and *semiblind deconvolution* [21]. As with normal deconvolution, b is the blurry image and z the blur-free image. In blind deconvolution, the PSF is entirely unknown and represented by y, while in semiblind deconvolution, information about the PSF is available (a parametric model, for example) and y represents a set of parameters that determine the PSF. (The precise meanings of these terms vary in the literature.) An actively researched application of blind deconvolution is the restoration of images corrupted by camera shake, especially since the work of [38, 71]. In this case the PSFs tend to be irregular and difficult to model, and information about them must be extracted almost entirely from the blurry images alone. Semiblind deconvolution is more commonly used in scientific imaging problems where PSF models are available, such as astronomical imaging with phase diversity [106].

1.3 Extreme ultraviolet solar imaging and the stray light problem

Much of this thesis was motivated by a single major project: a semiblind deconvolution problem involving extreme ultraviolet (EUV) images of the Sun. EUV images are used to study the solar corona, which reaches temperatures in excess of 1 megakelvin (MK), hundreds of times hotter than the underlying photosphere. The source of coronal heating is still not fully understood [48], and it is hoped that more and better solar observations will shed light on this question. EUV radiation spans wavelengths from $100 - 1000$ Ångströms (Å) and is generated only by extremely hot, ionized plasma. As a result the solar corona naturally emits EUV radiation, while the photosphere does not. NASA's fleet of spaceborne EUV telescopes - SOHO/EIT, TRACE, STEREO/EUVI, and SDO/AIA - has been observing the corona continuously for nearly two decades. The EUV corona seethes with activity and contains *loop arcades*, *coronal holes*, *filaments*, *plumes*, and other exotic structures (Fig. 1.1).

Coronal holes generate the fast solar wind [65], while filaments play an important role in coronal mass ejections, which erupt from the corona unpredictably and can cause expensive havoc in electrical systems on Earth [3].

EUV images carry a great deal of information about the coronal plasma. The plasma emissivity at a given position in the corona is roughly proportional to the square of the plasma density, while the temperature largely determines the radiation spectrum. Inverting this relationship, the plasma temperature, density, and other *diagnostics* can be determined from intensity measurements at multiple EUV wavelengths. This is typically done through the formalism of differential emission measure (DEM) analysis [52]. These data provide empirical constraints on the corona's behavior which help to determine the processes heating the corona and generating the solar wind.

A major difficulty in determining plasma diagnostics from EUV intensities is that a given pixel's intensity does not necessarily come from a small, uniform parcel of plasma. Instead, it is the sum of emissions in a tube around that pixel's geometric line of sight through the corona, a tube which may contain plasma of various temperatures and densities. Tomographic techniques can be used to create global 3D reconstructions of the corona free from line-of-sight contamination [39].

A common assumption of DEM, tomography, and other quantitative EUV analysis techniques is that solar radiation enters an EUV telescope and follows the ideal path predicted by geometric optics. In reality, light scatters off the geometric path and casts a haze of *stray light* over the image. Images of faint structures in the corona tend to have the highest stray light contamination. In particular, coronal holes, filament cavities, and structures far off the limb may be heavily contaminated, because they are much fainter than their surroundings [94]. Stray light can be corrected by deconvolution with the telescope PSF, which can be estimated by semiblind deconvolution of informative images. This deconvolution problem is unusual in that quantitative accuracy of the EUV intensities is the primary objective, not resolution or visual appeal.

1.4 Contributions of this thesis

In Chapter 2 we propose a generalization of variable elimination for solving separable inverse problems beyond least squares. Variable elimination is an optimization technique for (1.4) in which the optimal value of z in $F(y, z)$ is expressed as a function $z_m(y)$, and the reduced cost function $F(y, z_m(y))$ is optimized in place of $F(y, z)$. This

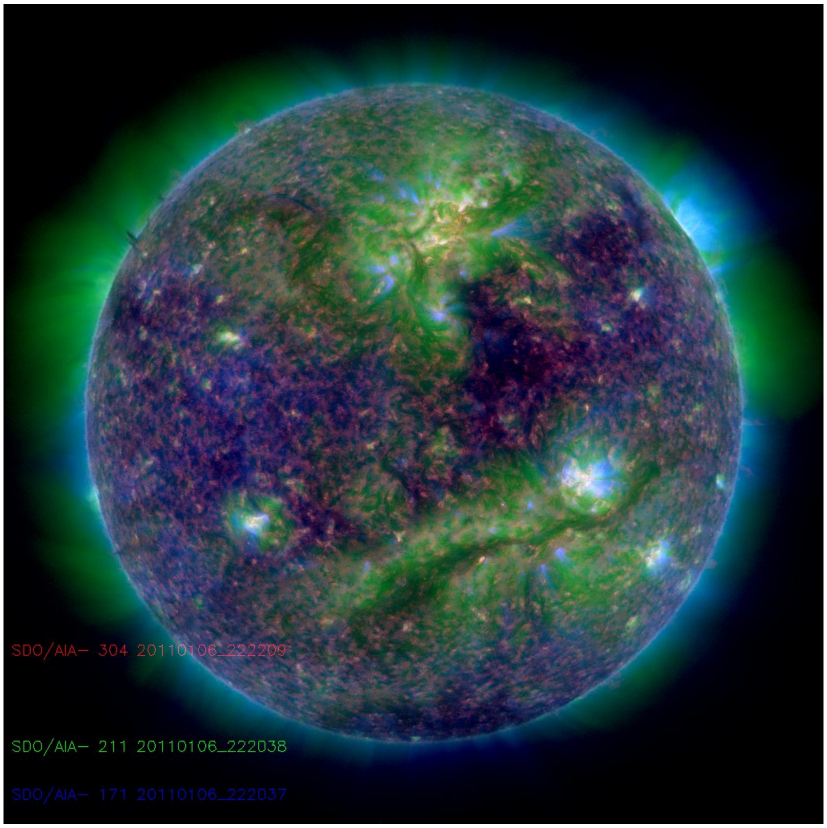
SDO/AIA- 304 20110106_222209
SDO/AIA- 211 20110106_222038
SDO/AIA- 171 20110106_222037

Figure 1.1: A composite of extreme ultraviolet images taken by SDO/AIA in three wavelengths: 171, 211, and 304 Å.

technique appears most prominently in the variable projection method of Golub and Pereyra [47]. Existing variable elimination methods require an explicit formula for the optimal value of the linear variables, so they cannot be used in problems with Poisson likelihoods, bound constraints, or other important departures from least squares. To address this limitation, we propose a generalization of variable elimination in which standard optimization methods are modified to behave as though a variable has been eliminated. We verify that this approach is a proper generalization by using it to rederive several existing variable elimination techniques. We then extend the approach to bound-constrained and Poissonian problems, showing in the process that many of the best features of variable elimination methods can be duplicated in our framework. Tests on difficult exponential sum fitting and blind deconvolution problems indicate that the proposed approach can have significant speed and robustness advantages over standard methods.

In Chapter 3 we propose a new method for correcting camera shake based on incremental sparse approximation of edges. The method first estimates the PSF, then uses non-blind deconvolution to obtain the sharp image. The PSF is estimated by solving an 'edge space' blind deconvolution problem: b represents the vertical and horizontal differences in the original blurry image, and z the same differences in the sharp image. An initial guess for the PSF is obtained by solving (1.4) with the constraint that z be much sparser than b, and the sparsity constraint is gradually relaxed to refine the PSF. A simple alternating projected gradient algorithm is used to perform the optimization. This simple method is shown to compete in deblurring performance with more sophisticated variational Bayes methods on a standard test set, while being significantly faster.

In Chapter 4 we determine PSFs that enable correction of stray light in solar images from all filter bands (171, 195, 284, and 304 Å) of the EUVI instruments aboard the STEREO-A and B spacecraft. Semi-empirical parametric formulas are proposed for the PSFs, and their parameters are determined by semiblind deconvolution of EUVI images. The EUVI-B PSFs were determined from lunar transit data, exploiting the fact that the Moon is not a significant EUV source. The EUVI-A PSFs were determined by analysis of simultaneous A/B observations from December 2006, when the instruments had nearly identical lines of sight to the Sun. We provide the first estimates of systematic error in EUV deconvolved images.

CHAPTER 2

A Generalization of Variable Elimination for Separable Inverse Problems Beyond Least Squares

2.1 Introduction

In linear inverse problems we are given a vector of noisy data $b \in \mathbb{R}^m$ generated by the linear model $b = Az + \epsilon$, where $A \in \mathbb{R}^{m \times c}$ is a known matrix, ϵ is a zero mean noise vector, and $z \in \mathbb{R}^{N_z}$ is an unknown vector with $N_z = c$ entries we wish to estimate. In separable inverse problems, A is not known exactly, but depends on another set of parameters $y \in \mathbb{R}^{N_y}$:

$$b = A(y)z + \epsilon. \tag{2.1}$$

The problem is now to determine the full set of $N = N_y + N_z$ parameters $x \triangleq (y, z)$.

Many scientific inverse problems are separable. In time-resolved spectroscopy and physical chemistry, data are often modeled as a weighted sum of several (possibly complex) exponentials with unknown decay rates [79,111]. Determining the weights and decay rates simultaneously is a separable inverse problem. Other examples include image deblurring with an incompletely known blur kernel [21] and tomographic reconstruction from incomplete geometric information [26]. Many more examples can be found in [45,53,87].

Separable problems frequently have additional exploitable structure. In this paper, we will be particularly interested in problems with multiple measurement vectors generated by applying a single linear transformation to n different vectors of linear coefficients. In this case, the data and coefficient vectors can be represented by

A version of this chapter has been published in the journal Inverse Problems with Anna C. Gilbert as co-author [95].

matrices $B \in \mathbb{R}^{m \times n}$ and $Z \in \mathbb{R}^{c \times n}$, and we have

$$B = A(y)Z + E. \tag{2.2}$$

This problem, also known as a *multiple right-hand sides* or *multi-way data* problem [46, 61, 78], occurs when a system is repeatedly observed under varying experimental conditions [79].

An inverse problem is generally solved by seeking parameter values that balance goodness of fit with conformity to prior expectations. In this paper we focus on constrained maximum likelihood problems, where we choose a goodness of fit function $L(A(y)z)$ measuring discrepancy between $A(y)z$ and b and a set $\mathcal{X} = \mathcal{Y} \times \mathcal{Z}$ representing known constraints on y and z, such as nonnegativity. We seek the parameter values that minimize the discrepancy subject to the constraints by solving

$$\min_{y \in \mathcal{Y}\, z \in \mathcal{Z}} \left\{ F(y, z) \triangleq L(A(y)z) \right\}. \tag{2.3}$$

Penalty functions such as ℓ_p norms on y and z may also be incorporated into $F(y, z)$, and while our techniques are relevant to this case, it is not specifically addressed here. For the goodness of fit function we use the negative log-likelihood $L(\mu) = -\log p(b \mid \mu)$, where the likelihood function $p(b \mid \mu)$ is the probability that $b = \mu + \epsilon$ and is determined by the distribution of ϵ. Least squares problems result from assuming standard Gaussian distributed noise, so that $L(\mu) = \frac{1}{2}\|\mu - b\|^2$, but Poissonian and other likelihoods frequently arise.

Unconstrained least squares problems are generally easiest to solve, and many powerful optimization ideas were first developed for this case [82]. However, unconstrained least squares solutions are not always satisfactory, and much better solutions can often be found using nonnegativity constraints, Poisson likelihoods, or other departures from ordinary least squares. Many physical quantities must be nonnegative, and enforcing this constraint can reduce reconstruction error [8] and help the optimizer avoid unphysical answers [96]. A Poisson process is often the best model for a stream of particles entering a detector, and in the low-count limit the Poisson and Gaussian distributions are very different. In this case Poissonian optimization usually gives significantly better parameter estimates than least squares, a fact of fundamental importance in astronomy [8, 9, 105], analytical chemistry [77], and biochemistry [68, 69], where information must be extracted efficiently from a trickle of incoming photons. This paper is concerned with advancing the state of the art for problems beyond least squares.

2.1.1 Existing optimization methods

We will focus on optimization methods employing *Newton-type* iterations. While other powerful methods exist for inverse problems, Newton-type methods enjoy very general applicability, attractive convergence properties, scalability under favorable conditions, and robustness against ill-conditioning and nonconvexity [82]. Given a smooth function $f(u)$, a constraint set $\mathcal{U} \subset \mathbb{R}^{N_u}$, and an initial point $u^0 \in \mathcal{U}$, a Newton-type method generates a sequence of iterates $u^1, u^2, \ldots$ which hopefully converge to the minimizer of $f(u)$ in $\mathcal{U}$, or at least a stationary point. Line search methods, which will be the focus of this paper, generally use the following update procedure to go from u^k to u^{k+1} [62, 82]:

1. *Search direction:* A search direction Δu is calculated by solving a Newton-type system of the form $B\Delta u = -g$, where g is determined from the gradient $\nabla f(u^k)$ and B is a *Hessian model* approximating $\nabla^2 f(u^k)$. Both g and B may be modified by information from constraints and previous iterates.

2. *Trial point calculation:* The step Δu determines a search path $u_p(s)$, parametrized by a step size $s > 0$, from which a trial point $\bar{u}$ is selected. This is generally a straight-line path modified to maintain feasibility with respect to constraints or hedge against a bad search direction.

3. *Evaluation and decision:* If moving to the trial point produces a sufficient decrease in the objective, we set $u^{k+1} = \bar{u}$. Otherwise, another trial point is constructed, possibly along a new direction Δu, and the process is repeated.

This update procedure is used in the service of some larger strategy for optimizing $F(y, z)$. To understand the strategies typically used, it is helpful to first consider strategies for solving the block-structured system $B\Delta x = -g$. This system has the block expansion

$$\begin{bmatrix} B_{yy} & B_{yz} \\ B_{zy} & B_{zz} \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} g_y \\ g_z \end{bmatrix}, \tag{2.4}$$

and is typically solved in one of three ways. (In the following, the product $M^{-1}w$ should be interpreted as a directive to solve $Mv = w$ for v rather than to compute M^{-1} explicitly, and when we speak of inversion we refer to this directive.)

1. *Full matrix, all-at-once.* We solve the whole system at once by QR or Cholesky factorization in medium-scale problems, and by conjugate gradients (CG) in very large-scale problems.

2. *Block Gauss-Seidel.* We converge to a solution by iterative updates of the form

$$\Delta y^{j+1} = -B_{yy}^{-1}(g_y - B_{yz}\Delta z^j) \tag{2.5}$$

$$\Delta z^{j+1} = -B_{zz}^{-1}(g_z - B_{zy}\Delta y^j). \tag{2.6}$$

   Gauss-Seidel is fast provided that B_{yy} and B_{zz} are much easier to invert than all of B and a block diagonal approximation of B is reasonably accurate, but may be arbitrarily slow to converge otherwise [91].

3. *Block Gaussian elimination.* By solving for Δz in the bottom row of (2.4) and substituting the result into the top row equation, we decompose (2.4) as

$$B_s\Delta y = -g_y + B_{yz}B_{zz}^{-1}g_z \tag{2.7a}$$

$$B_{zz}\Delta z = -g_z - B_{zy}\Delta y, \tag{2.7b}$$

   where $B_s \triangleq B_{yy} - B_{yz}B_{zz}^{-1}B_{zy}$ is the Schur complement of B_{zz} in B [16]. We construct the matrix B_s explicitly, solve for Δy in (2.7a), then plug the result into (2.7b) to solve for Δz.

Assuming B is positive definite, all three of these linear solvers can be interpreted as a method for minimizing the quadratic form $\frac{1}{2}\Delta x^T B\Delta x + g^T\Delta x$. Each of them can also be generalized to an update strategy for the nonquadratic problem (2.3), as follows:

1. *Full update:* We update y and z simultaneously using a step derived from solving the full system (2.4). Any classical Newton-type method applied directly to $F(y, z)$ falls into this category [82].

2. *Alternating update:* We make one or more updates to z with y fixed, then to y with z fixed, alternating until convergence [11]. Alternating methods now have well-developed convergence theory even with inexact alternating minimizations, and their iterations do not necessarily require matrix factorizations [15, 51]. As such, they may be the only tractable choice for certain large-scale and highly non-parametric problems such as nonnegative matrix factorization. However,

like Gauss-Seidel, alternating methods can converge slowly [18,90] and are generally preferable only when full updates are computationally expensive or intractable. In this paper we will focus on problems where full update methods are tractable, so alternation will not be considered further.

3. *Reduced update:* We determine the optimal z value given y,

$$z_m(y) = \underset{z\in\mathcal{Z}}{\operatorname{argmin}}\ F(y, z), \tag{2.8}$$

and substitute it into (2.3), giving an equivalent reduced problem

$$\min_{y\in\mathcal{Y}} \left\{ F_r(y) \triangleq F(y, z_m(y)) \right\}. \tag{2.9}$$

The Newton-type iteration is then applied to solve this reduced problem instead of the original. The resulting update has a nested structure: an outer optimizer computes the search direction Δy and trial point $\bar{y}$ to optimize $F_r(y)$, while an inner optimizer calculates z by solving (2.8) whenever the outer one asks for the value of $F_r(y)$ or its derivatives.

Most reduced update methods are variations on the variable projection algorithm of Golub and Pereyra [45, 47], which applies to the case of unconstrained separable least squares. In this case we have $F(y, z) = \frac{1}{2}\|A(y)z - b\|^2$ and $z_m(y) = A(y)^\dagger b$, where $X^\dagger$ denotes the Moore-Penrose pseudoinverse. Substituting $z_m(y)$ into $F(y, z)$ yields $F_r(y) = \frac{1}{2}\| - P_A^\perp b\|^2$, where $P_X^\perp = I - XX^\dagger$ denotes the projection onto range$(X)^\perp$, and the y in $A(y)$ has been suppressed. Golub and Pereyra proposed using a Gauss-Newton method to optimize $F_r(y)$. The Gauss-Newton method requires the Jacobian for the reduced residual $-P_A^\perp b$, which they derived by differentiation of pseudoinverses. This idea can also be extended to accommodate linear constraints on z.

The efficiency of variable projection in highly ill-conditioned curve fitting and statistical inference problems is theoretically and empirically well-attested [45, 84, 90, 97]. Variable projection is also useful for problems with multiple measurement vectors [46, 61, 78], as in this case $A(y)$ is block diagonal, so necessary pseudoinverses and derivatives may be efficiently computed blockwise. Other methods based on variable elimination can speed up the solution of large-scale image and volume reconstruction problems if the pseudoinverse and derivatives can be computed quickly [26,37,44,106].

Given the efficiency of variable elimination methods in separable least squares problems, one might hope to derive an extension with similar advantages to problems beyond least squares. However, such an extension runs into several difficulties. First, in problems beyond least squares there is generally no analytical formula for $z_m(y)$, and computing it is often computationally expensive. Second, if inequality constraints or nonsmooth penalties are imposed on z, then $z_m(y)$ will be a nonsmooth function with unpredictable properties, so that the reduced problem may be even more difficult than the original. Third, without a formula for $z_m(y)$ it is unclear how to compute $Dz_m(y)$, which is needed for a fast-converging second-order method.

2.1.2 Our contribution

Variable elimination does not seem to generalize easily to non-quadratic and constrained problems, but there are many efficient and robust full update methods for such problems [82]. This fact suggests that we might arrive at a generalization more easily from the other direction, by making existing full update methods resemble reduced update methods more closely. In this paper we explore the resulting *semi-reduced* update methods, explain how they relate to full and reduced update methods, describe when they are useful, and validate our claims with numerical experiments on hard inverse problems similar to ones encountered in practice.

In §2.2 we show how to transform a given full update method into a reduced method without an explicit formula for $z_m(y)$. We begin by applying two specific changes to the full update method: first, use block Gaussian elimination instead of an all-at-once solver, and second, adjust every new trial point's z coordinate to a better value before the trial point is evaluated. This second technique, which we call *block trial point adjustment*, is depicted graphically in Fig. 2.1, *right*. We call a full update method thus modified a semi-reduced method. Reduced methods are obtained from semi-reduced methods by requiring that the adjustment be optimal, which enables us to simplify the method by omitting computations of $\nabla_z F$ and the search direction Δz. We show reduced Newton and variable projection methods can be derived in this way. In §2.3, we propose and prove convergence of a semi-reduced method that allows for nonquadratic likelihoods and bound constraints on z, which has been posed as an open problem by multiple authors [26, 79].

The description of reduced and semi-reduced methods as modifications of full update methods allows one to predict when the former have advantages over the latter. Block Gaussian elimination is most effective when B_{zz} is easier to invert than all of B, for example when B_{zz} has block diagonal (Fig. 2.1), Toeplitz, banded, or other efficiently invertible structure. Block trial point adjustment should yield an efficiency gain when the computational burden of the adjustment subproblems is outweighed by an increase in convergence rate. This may occur when the graph of the objective contains a narrow, curved valley like that shown in Fig. 2.1.

To test these predictions we select problems where we expect semi-reduced methods to have an advantage, design methods for these problems using the semi-reduced framework, then compare the semi-reduced methods to standard full update methods. In §2.4 we derive linear algebra techniques that use block Gaussian elimination to exploit block structure or spectral properties of B, and in §2.5.1 and §2.5.2 we study two problems of scientific interest where these techniques have advantages over standard full-matrix methods. In §2.5.3 we consider a toy blind deconvolution problem where block trial point adjustment leads to a significant increase in convergence rate due to a curved valley geometry. We conclude that semi-reduced methods can have significant advantages over full update methods under the predicted conditions.

2.1.3 Related work

While the relationship between full and reduced update methods has been explored several times, the relationship established here is a major extension of previous work. In [90] Ruhe and Wedin developed the connection between full and reduced update Newton and Gauss-Newton methods, and semi-reduced methods are described by Smyth as *partial Gauss-Seidel* or *nested* methods in [97]. Our work extends theirs in that we consider general Newton-type methods, nonquadratic likelihoods, and the effect of globalization strategies, such as line search or trust regions, which ensure convergence to a stationary point from arbitrary initialization. A very general theoretical analysis of the relationship between the full and reduced problems is given in [86], but there is little discussion of practical algorithms and no mention of semi-reduced methods.

Structured linear algebra techniques such as block Gaussian elimination are known to be useful [22,113], but they are underutilized in practice. This is apparent from the fact that most optimization codes employ a limited set of broadly applicable linear algebra techniques [82], and very few are designed to accommodate user-defined linear solvers such as the ones we propose in §2.4. We contend that significant speed gains are attainable with special linear solvers, and optimization algorithm implementations should accommodate user-customized linear algebra by adding appropriate callback and reverse communication protocols.

Trial point adjustment is a key idea in the two-step line search and trust region algorithms of [29] and [30]. General convergence results are proven in [1] for 'accelerated' line search and trust region methods employing trial point adjustment. These

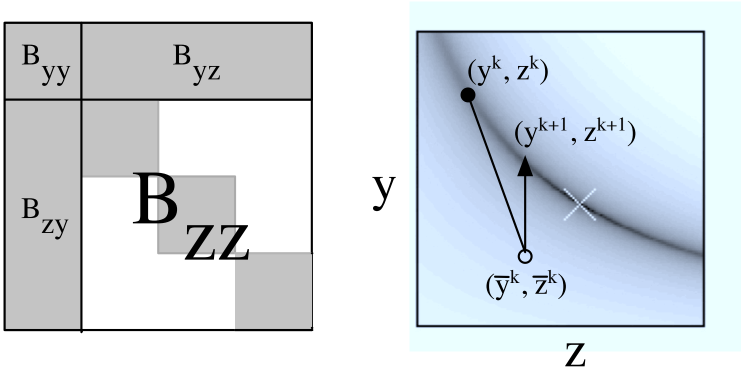

Figure 2.1: Situations where block gaussian elimination and trial point adjustment may be useful. *Left:* A 'block arrow' matrix B containing a block diagonal submatrix B_{zz} is well-suited for inversion by block Gaussian elimination. This type of matrix arises in separable problems with multiple measurement vectors. *Right:* Graph of an objective $F(y, z)$ exhibiting a narrow, curved valley; the minimum is marked with an X. Superimposed are a sample iterate (y^k, z^k) and an initial trial point $(\bar{y}^k, \bar{z}^k)$ that fails a sufficient decrease test. By adjusting this point's z coordinate to the minimum of $F(\bar{y}^k, z)$, we obtain a new trial point (y^{k+1}, z^{k+1}) that provides sufficient decrease to be accepted as an update.

works are not concerned with separable inverse problems or the relationship with reduced methods.

Extensions of variable projection beyond unconstrained least squares have been proposed, in particular to accommodate bound constraints on z [32, 96]. Their approach is to apply a Newton-type method to minimize $\tilde{F}_r(y) = F(y, \tilde{z}_m(y))$, an approximation of $F_r(y) = F(y, z_m(y))$ obtained by computing $z_m(y)$ approximately using a projected gradient or active set method. This approach can work well, but it has several theoretical and practical downsides. First, it has not been extended to nonquadratic likelihoods; second, computing $z_m(y)$ can be very expensive, and the precision required is unclear; third, an appropriate Hessian model is not obvious and must be obtained by ad-hoc heuristics or finite differences; and fourth, there has been no attempt at global convergence results. In contrast, our approach works on nonquadratic likelihoods; it provides the option of approximating $z_m(y)$ to any desired precision without danger of sacrificing convergence; one may use the same standard Hessian models used in full update methods, with exact derivatives if desired; and we prove a global convergence result for our method.

2.2 Semi-reduced methods

In this section we show that a full update algorithm may be transformed into a reduced update (variable elimination) algorithm by introducing block Gaussian elimination and an optimal block trial point adjustment, then simplifying the resulting algorithm to remove unnecessary computation. Semi-reduced methods are those obtained halfway through this process, after the two block techniques are introduced but before the simplification. We will describe the transformation process for unconstrained Newton-type line search algorithms, but it can be done with other types of algorithms too.

2.2.1 Simplification in the case of optimal adjustment

We begin the move towards semi-reduced methods by defining a standard unconstrained line search algorithm, then adding trial point adjustment. Let $f(u)$ be a twice-differentiable function and $\mathscr{B}(u) \in \mathbb{R}^{N_u \times N_u}$ the *Hessian model*, a positive definite matrix-valued function approximating $\nabla^2 f(u)$.

Given an iterate u^k, we obtain the update u^{k+1} by the following procedure. We begin by setting $g = \nabla f(u^k)$, $B = \mathscr{B}(u^k)$, and determining the search direction Δu by solving $B\Delta u = -g$. The search direction determines a line $u_p(s) = u^k + s\Delta u$ of potential trial points parametrized by step size s, and we set u^{k+1} by choosing one that satisfies the *sufficient decrease* condition

$$f(u_p(s)) - f(u^k) \leq \delta g^T(u_p(s) - u^k) = \delta g^T(s\Delta u), \tag{2.10}$$

for a fixed $\delta \in (0, 1/2)$. One can generally ensure convergence by picking a step size that obeys this condition and is not too small. Such a step size can be obtained by backtracking: we set $s = \alpha^j$ and try $j = 0, 1, 2, \ldots$ until (2.10) is satisfied.

To incorporate trial point adjustment into this update procedure, we assume we are given an *adjustment operator* $u_d(u)$ such that $f(u_d(u)) \leq f(u)$ for any input u. We then replace $u_p(s)$ with $u_d(u_p(s))$ on the left hand side of (2.10), obtaining Alg. 2.1. (Note that the standard full update method may be recovered by setting $u_d(u) = u$.) Global convergence of Alg. 2.1 to a stationary point is guaranteed by the following theorem:

Theorem 2.1. Assume that $f(u)$ is bounded below, $\nabla f(u)$ is Lipschitz continuous with bounded Lipschitz constant, the set $\{u \in \mathbb{R}^{N_u} \mid f(u) \leq f(u^0)\}$ is compact, and the matrices $\mathscr{B}(u)$ are symmetric positive definite with eigenvalues bounded away from zero and infinity. Then

$$\lim_{n\to\infty} \nabla f(u^k) = 0, \tag{2.11}$$

and any limit point of $(u^k)_{k=0}^{\infty}$ is a stationary point.

This theorem is proven in [62] for the standard algorithm without trial point adjustment, while an extension for algorithms including trial point adjustment is given in [1]. The line search condition in [1] is slightly different than the one used here, but the convergence argument applies without modification to our case. Informally, trial point adjustment does not harm convergence because convergence requires only that $f(u^k)$ decreases by some minimal amount for each iteration k, and the adjustment operator can only make the decrease larger.

Algorithm 2.1 Backtracking line search method with trial point adjustment

```
Require: $u^0 \in \mathbb{R}^{N_u}$, $\delta \in (0, 1/2)$, $\alpha \in (0, 1)$
for $k = 0, 1, 2, \ldots$ do
    $g = \nabla f(u^k)$, $B = \mathscr{B}(u^k)$
    Solve for $\Delta u$: $B\Delta u = -g$
    $u_p(s) = u^k + s\Delta u$
    Find the smallest $j \geq 0$ such that $f(u_d(u_p(\alpha^j))) - f(u^k) \leq \delta g^T(u_p(\alpha^j) - u^k)$
    $u^{k+1} = u_d(u_p(\alpha^j))$
end for
```

To make Alg. 2.1 into a semi-reduced method for minimizing a function $F(x) = F(y, z)$, we set $f(u) = F(x)$ and put system $B\Delta x = -g$ into the block Gaussian decomposed form (2.7). We then require the trial point adjustment to have the form $x_d(y, z) = (y, z_d(y, z))$, so that only z can change. The result of these changes is Alg. 2.2.

Algorithm 2.2 Semi-reduced line search method.

```
Require: $x^0 = (y^0, z^0) \in \mathbb{R}^N$, $\delta \in (0, 1/2)$, $\alpha \in (0, 1)$
Define $x_d(y, z) = (y, z_d(y))$
for $k = 0, 1, 2, \ldots$ do
    $g = \nabla F(x^k), B = \mathscr{B}(x^k)$
    Solve for $\Delta y$: $B_s \Delta y = -g_y + B_{yz} B_{zz}^{-1} g_z$
    Solve for $\Delta z$: $B_{zz} \Delta z = -g_z - B_{zy} \Delta y$
    Define $x_p(s) = (y_p(s), z_p(s)) = (y^k + s\Delta y, z^k + s\Delta z)$
    Find the smallest $j \geq 0$ such that $F(x_d(x_p(\alpha^j))) - F(x^k) \leq \delta g^T(x_p(\alpha^j) - x^k)$
    $x^{k+1} = x_d(x_p(\alpha^j))$
end for
```

To make Alg. 2.2 into a reduced update method, we assume our trial point adjustment is unique and optimal, $z_d(y,z) = z_m(y) = \text{argmin}_z\, F(y,z)$, and exploit this fact to simplify the algorithm. Optimal adjustments ensure that $g_z = \nabla_z F(y^k, z_m(y^k)) = 0$ for all k, so terms involving g_z disappear. In particular, line 7 reduces to $g^T(x_p(\alpha^j) - x^k) = g_y^T(y_p(\alpha^j) - y^k)$. After terms involving g_z are removed, the trial point $z_p(s) = z^k + s\Delta z$ appears only within the expression $x_d(x_p(\alpha^j))$. But if we write out $x_d(x_p(s)) = (y^k + s\Delta y, z_m(y^k + s\Delta y))$, we see that $z^k + s\Delta z$ has been supplanted by the adjusted point $z_m(y^k + s\Delta y)$, so we may skip it by redefining x_p as $x_p(s) = (y^k + s\Delta y, z_m(y^k + s\Delta y))$. The disappearance of $z^k + s\Delta z$ renders the step Δz unused in any way, so line 5 can be deleted. What is left is Alg. 2.3, a *simplified semi-reduced method*. In the next section we show that, when B is chosen appropriately, versions of this simplified semi-reduced method are identical to several reduced (variable elimination) methods in the literature.

Algorithm 2.3 Simplified semi-reduced line search method.

```
Require: $x^0 = (y^0, z_m(y^0)) \in \mathbb{R}^N$, $\delta \in (0, 1/2)$, $\alpha \in (0,1)$
for $k = 0, 1, 2, \ldots$ do
    $g_y = \nabla_y F(x^k), B = \mathscr{B}(x^k)$
    Solve $B_s \Delta y = -g_y$
    Define $x_p(s) = (y_p(s), z_p(s)) = (y^k + s\Delta y, z_m(y^k + s\Delta y))$
    Find the smallest $j \geq 0$ such that $F(x_p(\alpha^j)) - F(x^k) \leq \delta g_y^T (y_p(\alpha^j) - y^k)$
    $x^{k+1} = x_p(\alpha^j)$
end for
```

Reinterpreting variable elimination as a simplified semi-reduced method allows us to precisely articulate the cost-benefit tradeoff involved in using variable elimination, as well as the *raison d'être* for *non*-simplified semi-reduced methods. The benefit of variable elimination is that we need not compute g_z, Δz, or quantities dependent on them, and the trial point adjustments may cause the algorithm to converge faster. The cost is that we must compute the *optimal* z value after every y update, while in semi-reduced updates we only require that the adjustment does not increase the objective. Variable elimination is preferable only if the adjustment subproblem can be solved quite quickly and yields a significantly increased convergence rate. While this condition often holds in unconstrained least squares problems, in general calculating $\text{argmin}_z\, F(y,z)$ is often quite costly and may not be worth the trouble. Semi-reduced methods permit us to forgo this cost, granting increased flexibility without compromising convergence.

2.2.2 Equivalence of simplified semi-reduced methods to variable elimination

Here we show that three popular reduced (variable elimination) methods can all be interpreted as simplified semi-reduced methods with an appropriate Hessian model. In other words, reduced methods can be obtained by operations on $F(y, z)$ alone, without ever forming the objective $F_r(y)$ explicitly. This surprising result is essentially due to the implicit function theorem and the fact that optimization methods only use very limited local information about a function to determine iterates. We begin with a new lemma stating the exact condition required for a reduced and a simplified semi-reduced method to be equivalent.

Lemma 2.1. Let $y^0 \in \mathbb{R}^{N_y}$ be given, and let $z^0 = z_m(y^0)$. Let invertible Hessian models $\mathscr{B}_r(y)$ and $\mathscr{B}_f(y, z)$ for $F_r(y)$ and $F(y, z)$ be given. Assume that $z_m(y)$ is well-defined: that is, there is a unique solution of $\min_z F(y, z)$ for any given y. Consider the following pair of Newton-type algorithms:

1. *Reduced method:* Alg. 2.1 with $f(u) = F_r(y)$, $y_d(y) = y$, $\mathscr{B} = \mathscr{B}_r$.

2. *Simplified semi-reduced method:* Alg. 2.3 with $\mathscr{B} = \mathscr{B}_f$.

Let $B_s = B_{yy} - B_{yz}B_{zz}^{-1}B_{zy}$. These two algorithms generate identical iterates if and only if, at all points y^k visited by each algorithm, the Hessian models $B_r = \mathscr{B}_r(y)$ and $B = \mathscr{B}_f(y, z_m(y))$ obey

$$B_r = B_s. \tag{2.12}$$

Proof. After the specified substitutions are made, Algs. 2.1 and 2.3 have exactly one difference: the gradient used in Alg. 2.1 is $\nabla F_r(y)$, while in Alg. 2.3 it is $\nabla_y F(y, z)$. Thus it suffices to show that $\nabla F_r(y) = \nabla_y F(y, z)$. Letting Dz_m denote the Jacobian of $z_m(y)$, we have

$$\nabla F_r(y) = \nabla_y F(y, z_m(y)) + Dz_m \cdot \nabla_z F(y, z_m(y)) = \nabla_y F(y, z_m(y)), \tag{2.13}$$

where the second term has vanished because $z_m(y)$ is a stationary point of $F(y, z)$, so $\nabla_z F(y, z_m(y)) = 0$. □

Now we show that the reduced Newton method (i.e. Newton's method on $F_r(y)$) can be interpreted as a simplified semi-reduced Newton method on $F(y, z)$. This was implicitly shown by Richards [89] for the classical, nonglobalized Newton iteration.

Proposition 2.2. Under the assumptions of Lemma 2.1, the reduced method Alg. 2.1 with Hessian model $B_r = \nabla^2 F_r$ is equivalent to the simplified semi-reduced method Alg. 2.3 with model $B = \nabla^2 F$.

Proof. We need only verify the Schur complement relation (2.12). Differentiating (2.13), we have

$$\nabla^2 F_r = \nabla^2_{yy} F + \nabla^2_{yz} F \cdot Dz_m. \tag{2.14}$$

Dz_m can be obtained by implicit differentiation of the stationary point condition $\nabla_z F(y, z_m(y)) = 0$:

$$\nabla^2_{zy} F(y, z_m(y)) + \nabla^2_{zz} F(y, z_m(y)) \cdot Dz_m = 0 \tag{2.15}$$

$$Dz_m = -[\nabla^2_{zz} F]^{-1} \nabla^2_{zy} F. \tag{2.16}$$

Plugging this expression into (2.14) and setting $B_r = \nabla^2 F_r$ and $B = \nabla^2 F$ yields (2.12) as desired. □

Now consider the separable case, where $F(y, z) = L(A(y)z)$, but $L(\mu)$ is not necessarily a least squares functional. We derive two simplified semi-reduced methods for this objective. In the least squares case, these methods are equivalent to the Kaufman [60] and Golub-Pereyra [45, 47] variants of variable projection, but they also apply to general nonquadratic L, a case for which no reduced method existed before. To derive our methods, we note that the variable projection model Hessians B_r have a *closed-form normal decomposition*: they can be written as $B_r = X_r^T X_r$ for some explicit X_r. Accordingly we will seek Hessian models B such that $B_s = X_s^T X_s$ for some closed-form X_s.

We set some notation and conventions before we begin. Let $X_{:,j}$ the j^{th} column of a matrix X. For any full column rank matrix X, $X^\dagger = (X^T X)^{-1} X^T$ is the Moore-Penrose psuedoinverse and $P_X^\perp = I - XX^\dagger$ is the orthogonal projector onto range$(X)^\perp$. Given a function $f(u, v)$ let $Df = [\partial_u f, \partial_v f]$ denote its Jacobian. To simplify our formulas we define the quantities $\mu(y, z) = A(y)z$, $W = (\nabla^2 L)_\mu^{1/2}$, and $\bar{A} = WA$. We abuse notation by ignoring the implicit dependence of W on y and z, which allows us to write $W\partial_{y_j} A$ as $\partial_{y_j} \bar{A}$.

We begin by decomposing the full Hessian of F into two components: $\nabla^2 F = G + E$. The G term is the Gauss-Newton Hessian model, $G = J^T J$, where $J = W(D\mu)$. The blocks of J are given by

$$(J_y)_{:,j} = (\partial_{y_j} \bar{A}) z \text{ for } j = 1, \ldots, N_y, \qquad J_z = \bar{A}. \tag{2.17}$$

The E component is a residual term given by $E = \sum_i (\nabla L)_i \nabla^2 \mu_i$. Note that $E_{zz} = 0$ because $\nabla^2_{zz}\mu_i = 0$ for all i.

The first Hessian model we consider will be G. A closed-form normal decomposition for G_s can be derived by:

$$G_s = J_y^T(I - \bar{A}\bar{A}^\dagger)J_y = J_y^T P_{\bar{A}}^\perp J_y = (-P_{\bar{A}}^\perp J_y)^T(-P_{\bar{A}}^\perp J_y) = J_s^T J_s, \tag{2.18}$$

where the last line uses the fact that orthogonal projection is symmetric and idempotent, and the minus sign has been introduced for consistency with the variable projection convention. By Lemma 2.1 this result yields a pair of equivalent reduced and simplified semi-reduced methods for any $L(\mu)$:

Proposition 2.3. The reduced method Alg. 2.1 with Hessian model $B_r = G_s$ is equivalent to the simplified semi-reduced method Alg. 2.3 with model $B = G$.

In the least squares case we have $z = z_m(y) = A^\dagger b$, so $J_s = -P_A^\perp(\partial_{y_j} A)z = -P_A^\perp(\partial_{y_j} A)A^\dagger b$, and this J_s is precisely the reduced Jacobian J_r proposed by Kaufman. Thus we have $G_s = G_r$ and the following result, which was proven by Ruhe and Wedin in [90] for algorithms without globalization:

Corollary 2.4. Kaufman's variable projection method is equivalent to a simplified semi-reduced method for separable least squares using $B = G$.

Next we express the Golub-Pereyra variable projection method as a simplified semi-reduced method. To do this we need a Hessian model H such that $H_s = K_s^T K_s$, where K_s is equal to the Golub-Pereyra reduced Jacobian K_r. This is a challenging problem because the Golub-Pereyra model $H_r = K_r^T K_r$ is a closer approximation to $\nabla^2 F_r$ than the Kaufman model G_r, but there is no obvious normally decomposable H that approximates $\nabla^2 F$ better than the traditional Gauss-Newton model G.

Fortunately the model may be derived by an ingenious technique due to Ruhe and Wedin. Essentially, their idea is to apply a block Cholesky factorization to $\nabla^2 F$ and use the factors to help reduce the discrepancy between G and $\nabla^2 F$. In our notation the Cholesky factorization used is the UDU^T factorization, which is simply the more familiar LDL^T factorization [82] with the conventional variable order reversed. Given a matrix X, we write its UDU^T factorization as $X = U\hat{X}U^T$, where

$$\hat{X} = \begin{bmatrix} X_s & 0 \\ 0 & X_{zz} \end{bmatrix}, \quad U = \begin{bmatrix} I & X_{yz}X_{zz}^{-1} \\ 0 & I \end{bmatrix} \tag{2.19}$$

and $X_s = X_{yy} - X_{yz}X_{zz}^{-1}X_{zy}$. Note that the U factor is determined uniquely by its yz block. Setting $X = \nabla^2 F$ we have $U_{yz} = (G_{yz} + E_{yz})G_{zz}^{-1}$.

To derive H, consider the product $U^{-1}GU^{-T}$, which is positive definite and normally decomposable because G is. If G were the true Hessian $U^{-1}GU^{-T}$ would be block diagonal, but in reality

$$U^{-1}GU^{-T} = \begin{bmatrix} G_s + E_{yz}G_{zz}^{-1}E_{zy} & -E_{yz} \\ -E_{zy} & G_{zz} \end{bmatrix}. \tag{2.20}$$

Letting $\hat{H}$ denote the block diagonal of $U^{-1}GU^{-T}$, we can define a new positive definite and normally decomposable Hessian model by setting $H \triangleq U\hat{H}U^T$. From (2.19) it immediately follows that

$$H_s = \hat{H}_{yy} = G_s + E_{yz}G_{zz}^{-1}E_{zy}. \tag{2.21}$$

Now we express H_s in the form $H_s = K_s^T K_s$. We have already decomposed $G_s = J_s^T J_s$; a similar formula for the second term, $E_{yz}G_{zz}^{-1}E_{zy}$, is given by

$$E_{yz}G_{zz}^{-1}E_{zy} = E_{yz}(\bar{A}^T\bar{A})^{-1}E_{zy} = [(\bar{A}^\dagger)^T E_{zy}]^T[(\bar{A}^\dagger)^T E_{zy}] = M^T M, \tag{2.22}$$

where we have used the identity $(X^T X)^{-1} = X^\dagger (X^\dagger)^T$ valid for any matrix X with full column rank. We now have $H_s = J_s^T J_s + M^T M$, where $J_s = -P_{\bar{A}}^\perp J_y$ and $M = (\bar{A}^\dagger)^T E_{zy}$. Surprisingly we may rewrite this as $H_s = (J_s + M)^T(J_s + M)$ because the cross terms vanish: $J_s^T M = -J_y^T P_{\bar{A}}^\perp (\bar{A}^\dagger)^T E_{zy} = 0$ for $P_X^\perp (X^\dagger)^T = 0$. Therefore, by setting $K_s = J_s + M$, we have $H_s = K_s^T K_s$ as desired.

All that remains is to compute K_s, which we do column-by-column. The j^{th} column of J_s is $(J_s)_{:,j} = (-P_{\bar{A}}^\perp J_y)_{:,j} = -P_{\bar{A}}^\perp(\partial_{y_j}\bar{A})z$, while the j^{th} column of E_{zy} is given elementwise by

$$(E_{zy})_{kj} = \sum_i (\nabla L)_i \partial_{z_k}\partial_{y_j}(Az)_i = \sum_i (\nabla L)_i (\partial_{y_k} A)_{ik} = [(\partial_{y_j} A)^T \nabla L]_k, \tag{2.23}$$

so we have $M_{:,j} = (\bar{A}^\dagger)^T(\partial_{y_j}A)^T \nabla L$. We write this in terms of $\bar{A}$ by defining the weighted residual $r = W^{-1}\nabla L$, so that $M_{:,j} = (\bar{A}^\dagger)^T(\partial_{y_j}\bar{A})^T r$. Thus the desired formula for K_s's columns is

$$(K_s)_{:,j} = (J_s)_{:,j} + M_{:,j} = -P_{\bar{A}}^\perp(\partial_{y_j}\bar{A})z + (\bar{A}^\dagger)^T(\partial_{y_j}\bar{A})^T r. \tag{2.24}$$

Again invoking Lemma 2.1, we have shown that

Proposition 2.5. The reduced method Alg. 2.1 with Hessian model $B_r = H_s$ is equivalent to the simplified semi-reduced method Alg. 2.3 with model $B = H$.

Specializing this result to the least-squares case $L(\mu) = \frac{1}{2}\|\mu - b\|^2$ as before, we have $r = Az - b = AA^{\dagger}b - b = -P_A^{\perp}b$, and $(K_s)_{:,j}$ simplifies to

$$(K_s)_{:,j} = -\left(P_A^{\perp}(\partial_{y_j}A)A^{\dagger} + (P_A^{\perp}(\partial_{y_j}A)A^{\dagger})^T\right)b, \tag{2.25}$$

which is precisely the Jacobian K_r of the reduced functional $F(y, z_m(y)) = \frac{1}{2}\|-P_A^{\perp}b\|^2$ derived by Golub and Pereyra [45]. Since $K_r = K_s$, we have $H_r = H_s$ and the desired equivalence:

Corollary 2.6. The Golub-Pereyra variable projection method is equivalent to a simplified semi-reduced method for separable least squares using Hessian model H.

2.2.3 Semi-reduced methods as the natural generalization of variable elimination

Proposition 2.2 and Corollaries 2.4 and 2.6 show that the reduced Newton's method and both variants of variable projection can be interpreted as simplified semi-reduced methods. In addition, Propositions 2.3 and 2.5 define new simplified semi-reduced methods that generalize variable projection to nonquadratic $L(\mu)$.

Unfortunately these algorithms are of more theoretical than practical use, for the following reasons. First, we still have not dealt with the problem of computing $z_m(y)$. In general there is no closed form for $z_m(y)$, and computing it may be so expensive that the computational burden outweighs any increase in convergence rate over a simple full or alternating update method. Second, if the domain of $\ell_i(\mu_i)$ is a bounded subset of $\mathbb{R}$, as is true for the Poisson and several other log-likelihoods, the bounds often must be enforced via reparametrization or constrained optimization. This adds still more complexity and in the latter case makes unconstrained optimization inapplicable.

The driving technical insight of this paper is the following: if we forgo the simplifications afforded by using optimal block trial point adjustment and use an ordinary semi-reduced method instead, *all of these barriers and difficulties disappear.* Trial point adjustments need not be optimal, so there is no need for the computationally expensive $z_m(y)$, and constraints can be handled by incorporating trial point adjustment and block Gaussian elimination into classical full update methods. Thus, semi-reduced methods provide a natural way to extend variable elimination methods beyond least squares.

2.3 A semi-reduced method for bound constrained and non-quadratic problems

To illustrate this point, in this section we present a classical method for smooth bound-constrained problems and turn it into a semi-reduced method. The problem we wish to solve is

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad l \leq x \leq u, \tag{2.26}$$

where $-\infty \leq l \leq u \leq \infty$ are vectors bounding the components of $x \in \mathbb{R}^N$, and $f(x)$ is twice differentiable. The method we present is a trial point adjusted variant of Bertsekas's projected Newton method [10, 41]. (In our terminology Bertsekas's method is better described as a *projected Newton-type method*, since it allows for approximate Hessian models.) We choose this method because it is relatively simple, its convergence is global and potentially superlinear, and similar second-order gradient projection methods are empirically among the state-of-the-art for a variety of constrained inverse problems [8, 92, 98, 105].

The update in the projected Newton-type method is of the form

$$x^{k+1} = \mathcal{P}(x^k - S^k \nabla f(x^k)), \tag{2.27}$$

where S^k is a scaling matrix which we will assume to be positive definite, and $\mathcal{P}(w)$ is the projection of w onto the box $\mathcal{B} = \{w \,|\, l \leq w \leq u\}$ given componentwise by

$$\mathcal{P}(w)_i \triangleq \text{median}(l_i, w_i, u_i). \tag{2.28}$$

This iteration is a generalization of the projected gradient method, which restricts S^k to be a multiple of the identity. Bertsekas showed that the naive Newton-type choice $S^k = (B^k)^{-1}$ with $B^k \approx \nabla^2 f(x^k)$ can cause convergence failures, but convergence can be assured by modifying the naive choice using a very simple active set strategy, in which the Hessian model B^k is modified to be diagonal with respect to the active indices.

To describe projected Newton-type methods we will use the following notation. Let $[N] \triangleq \{1, \ldots, N\}$, and for any $J \subset [N]$, let $v_J = (v_i)_{i \in J}$ denote the subvector of $v \in \mathbb{R}^N$ indexed by J and $X_{J,J} = [X_{ij}]_{i,j\in J}$ the indexed submatrix of $X \in \mathbb{R}^{n\times n}$. Given $\epsilon \geq 0$ and $x \in \mathbb{R}^n$, we define the active set associated with x by

$$\mathcal{A}(x) = \{i \in [N] \,|\, (\nabla f(x)_i > 0,\ x_i \leq l_i + \epsilon) \text{ or } (\nabla f(x)_i < 0,\ x_i \geq u_i - \epsilon)\}, \quad (2.29)$$

and the inactive set as its complement, $\mathcal{I}(x) = [N] - \mathcal{A}(x)$. Using this notation we present in Alg. 2.4 the projected Newton-type method with trial point adjustment, where an identity scaling matrix is chosen on the active set for concreteness. As in the unconstrained case, the only difference between the semi-reduced method Alg. 2.4 and the original full update method (found in equations (32)–(37) of [10]) is the addition of the adjustment operator $x_d(x)$: specifically, on the left hand side of line 5 and the right hand side of line 6 of Alg. 2.4, our method has $x_d(x_p(\alpha^j))$ while [10] has only $x_p(\alpha^j)$. Careful examination of the global convergence proof, Proposition 2 in [10], reveals that, with very minor additions, it also establishes convergence of Alg. 2.4. Here we review the argument very briefly, with just enough detail to describe how to adapt it to accommodate trial point adjustment.

Algorithm 2.4 Projected Newton-type method with trial point adjustment.

```
Require: $x^0 \in \mathbb{R}^N$, $\delta \in (0, 1/2)$, $\alpha \in (0, 1)$
1: for $k = 0, 1, 2, \ldots$ do
2:     $g = \nabla f(x^k)$, $B^k = \mathscr{B}(x^k)$, $A = \mathcal{A}(x^k)$, $I = \mathcal{I}(x^k)$
3:     $\Delta x_I = -(B^k_{I,I})^{-1} g_I$, $\Delta x_A = -g_A$
4:     Define $x_p(s) = \mathcal{P}(x^k + s\Delta x)$
5:     Find the smallest $j \geq 0$ such that
           $$f(x_d(x_p(\alpha^j))) - f(x^k) \leq \delta \left\{ g_I^T(\alpha^j \Delta x_I) + g_A^T(x_p(\alpha^j)_A - x^k_A) \right\} \quad (2.30)$$
6:     $x^{k+1} = x_d(x_p(\alpha^j))$
7:     $\epsilon \leftarrow \min(\epsilon_0, \|x_p(1) - x^k\|)$
8: end for
```

Proposition 2.1. Assume that $\nabla f(x)$ is Lipschitz continuous on any bounded set of $\mathbb{R}^N$ and the eigenvalues of B^k are uniformly bounded away from zero and infinity for all k. Then every limit point of Alg. 2.4 is a stationary point of Problem (2.26).

Proof. By contradiction: suppose a subsequence $(x^k)_{k\in K}$ of $(x^k)_{k=0}^{\infty}$ exists such that $\lim_{k\to\infty, k\in K} x^k = \bar{x}$, where $\bar{x}$ is not a critical point. Let $s_k = \alpha^{j_k}$ denote the step size chosen at iteration k on line 5; the proof of Proposition 2 of [10] first shows that the monotonicity of the sequence $(f(x^k))_{k=0}^{\infty}$, Lipschitz continuity of $\nabla f(x)$, eigenvalue bound on B^k, and the nonpositivity of the terms on the right hand side of (2.30) together imply that $\liminf_{k\to\infty, k\in K} s_k = 0$. Since all the required properties still hold in our case, this conclusion holds for Alg. 2.4 as well. Next it is shown that, for some $\bar{s} > 0$ independent of k, we have

$$f(x_p(s)) - f(x_k) \leq \delta\{g_I^T(s\Delta x_I) + g_A^T(x_p(s)_A - x_A^k)\} \quad \text{for } s \leq \bar{s}. \tag{2.31}$$

The computation supporting this claim depends only on the properties of $f(x)$, $x_p(s)$, A, and I, so it still holds for Alg. 2.4. But $f(x_d(x)) \leq f(x)$ for all x, so

$$f(x_d(x_p(s))) - f(x_k) \leq \delta\{g_I^T(s\Delta x_I) + g_A^T(x_p(s)_A - x_A^k)\} \quad \text{for } s \leq \bar{s}. \tag{2.32}$$

It follows that $s_k \geq \alpha^J$ where J is the smallest nonnegative integer such that $\alpha^J \leq \bar{s}$, contradicting $\liminf_{k\to\infty, k\in K} s_k = 0$. □

Careful examination of the proofs in [10] indicates that the other properties of the projected Newton-type method generally continue to hold for the trial point adjusted version, but the full details are beyond the scope of this paper.

Since B is only required to have eigenvalues bounded away from 0 and ∞, Alg. 2.4 can accommodate a wide variety of Hessian models and regularization strategies. In our numerical experiments we use Alg. 2.5, which is a special case of Alg. 2.4 and thus inherits its convergence properties. Alg. 2.5 sets $B = \mathscr{B}(x) + \lambda I$, where $\mathscr{B}(x)$ is a Gauss-Newton Hessian, λI is a Levenberg-Marquardt damping term, and λ is adjusted at every iteration according to a step quality metric ρ. Levenberg-Marquardt regularization is useful for guarding against rank-deficient Hessian models [82].

While any linear algebra technique may be used to solve the Newton-type systems in Algs. 2.4 and 2.5, block Gaussian elimination is of particular interest because of the role it plays in our semi-reduced framework. The block Gaussian elimination methods used in our numerical experiments are introduced in the next section.

2.4 Using block Gaussian elimination to exploit separable structure

One of the key advantages of variable elimination methods is their ability to take advantage of special problem structure, such as multiple measurement vectors [61]. Block Gaussian elimination can be used to derive linear solvers with similar structure exploiting properties. Here we describe two such algorithms which we claim can provide an advantage over standard methods; these claims are tested in our

Algorithm 2.5 Damped projected Newton-type method with trial point adjustment.

```
Require: $\delta \in (0, 1/2)$, $\alpha \in (0, 1)$, $\lambda_{min}, \lambda_{max} \in [0, \infty)$, $0 \le \rho_{bad} < \rho_{good} \le 1$
Require: $\tau \in (0, \infty)$, $\epsilon_0 \in (0, \infty)$
$\epsilon \leftarrow \epsilon_0$
for $k = 0, 1, 2, \ldots$ do
    $B = \mathscr{B}(x^k) + \lambda I$, $g = \nabla f(x^k)$, $A = \mathcal{A}(x^k)$, $I = \mathcal{I}(x^k)$
    If $\|\mathcal{P}(x^k - g) - x^k\| \le \tau$ or $k \ge k_{max}$, stop.
    Solve $B_{I,I}\Delta x_I = -g_I$. Set $\Delta x_A = -g_A$.
    Define $x_p(s) = \mathcal{P}(x^k + s\Delta x)$
    Find the smallest $j \ge 0$ such that
        $$f(x_d(x_p(\alpha^j))) - f(x^k) \le \delta\left\{g_I^T(\alpha^j \Delta x_I) + g_A^T(x_p(\alpha^j)_A - x_A^k)\right\}$$
    $x^{k+1} = x_d(x_p(\alpha^j))$
    $\rho = (f(x^k) - f(x^{k+1}))/(-\frac{1}{2}g_I^T\Delta x_I)$
    if $\rho > \rho_{good}$ then
        $\lambda \leftarrow \max(\lambda/2, \lambda_{min})$
    else if $\rho < \rho_{bad}$ then
        $\lambda \leftarrow \min(10\lambda, \lambda_{max})$
    end if
    $\epsilon \leftarrow \min(\epsilon_0, \|x_p(1) - x^k\|)$
end for
```

experiments below.

The first method is a QR method for normal equations, and is thus appropriate for methods employing a Gauss-Newton Hessian model. This approach is most suited for highly ill-conditioned systems, such as those arising from exponential fitting and other difficult problems traditionally tackled by variable projection. Generalized Gauss-Newton Hessian models for nonquadratic likelihoods can be handled by this method [19]. The second method is for problems where z is very high dimensional (a vectorized image or volume array for example), while y is relatively low-dimensional. It is similar to the linear algebra algorithms used in the reduced update optimizers defined in [26, 106]. Unlike these algorithms, which are designed for specific least squares optimization tasks, our algorithm can be used in any Newton-type optimizer, including ones that handle Poisson likelihoods or bound constraints.

2.4.1 Solving normal equations by block decomposed QR factorization

We now present a method for solving normal equations by block Gaussian elimination and QR factorization. Normal equations are systems of the form

$$J^T J \Delta x = -J^T r, \tag{2.33}$$

where $J \in \mathbb{R}^{m \times N}$. The Newton-type system $B\Delta x = -g$ has this form when we use a Gauss-Newton Hessian model or its generalization for non-quadratic likelihoods [19]. Assuming $B = J^T J$, the reduced and damped Gauss-Newton system $(B_{I,I} + \lambda I)\Delta x_I = -g_I$ from Alg. 2.5 can also be written in this form by deleting columns from J and augmenting the result with the scaled identity matrix $\sqrt{\lambda} I$ [82].

Cholesky factorization is the fastest way to solve normal equations, but rounding error can amplify to unacceptable levels when J is highly ill-conditioned, as in some curve fitting problems. Greater accuracy can be gained at the expense of additional computation by QR factorizing J. Assuming J is full rank, we will write the (thin) QR factorization as $[Q, R] = \texttt{qr}(J)$, where $Q \in \mathbb{R}^{m \times N}$ is an orthogonal matrix and $R \in \mathbb{R}^{N \times N}$ is an invertible upper triangular matrix. Substituting $J = QR$ into (2.33) and noting that $Q^T Q = I$, we obtain the solution

$$\Delta x = -R^{-1} Q^T r. \tag{2.34}$$

In our method we solve (2.33) by QR factorizing not the system itself, but its block decomposed form (2.7). We begin by putting (2.7) in normal equation form. From (2.18) we have $B_s = J_s^T J_s$, where $J_s = P_{J_z}^{\perp} J_y$. Similarly we have $-g_y + B_{yz} B_{zz}^{-1} g_z = -J_s^T r$. From these results we can write (2.7) as a pair of normal equations:

$$(J_s^T J_s)\Delta y = -J_s^T r \tag{2.35a}$$

$$(J_z^T J_z)\Delta z = -J_z^T (r + J_y \Delta y). \tag{2.35b}$$

To compute J_s, we need to compute $P_{J_z}^{\perp}$. This may be done using the QR factorization of J_z: if $X = QR$ is the QR factorization of a matrix X with full column rank, we have

$$P_X^{\perp} = I - QQ^T. \tag{2.36}$$

Using this result we can form J_s and solve the system as described in Alg. 2.6.

Alg. 2.6 is useful when J_z has structure that makes its QR factorization easier to compute than that of the full J. As an example, suppose that J_z is a block diagonal

Algorithm 2.6 Solution of $J^T J \Delta x = -J^T r$ by block decomposed QR.

```
[$Q_z, R_z$] = $\texttt{qr}(J_z)$
$[t, T] = Q_z^T [r, J_y]$
$J_s = J_y - Q_z T$
$[Q_s, R_s] = \texttt{qr}(J_s)$
$\Delta y = -R_s^{-1} Q_s^T r$
$\Delta z = -R_z^{-1}(t + T\Delta y)$
```

matrix with blocks $J_z^{(i)}$ for $i = 1, \ldots, n$. Such matrices arise in separable problems with multiple measurement vectors. In this case Q_z and R_z are block diagonal and Alg. 2.6 can be adapted to exploit this, as shown in Alg. 2.7. Note that this algorithm never generates the large sparse matrix J, but only the nonzero blocks $J_y^{(i)}$ and $J_z^{(i)}$, which are computed just when they are needed. We expect this resource economy to result in reduced memory usage, higher cache efficiency, and ultimately a faster solution.

Algorithm 2.7 Alg. 2.6 specialized to the case of block diagonal J_z.

```
for $i = 1, \ldots, n$ do
    Compute $J_y^{(i)}, J_z^{(i)}$
    $[Q_z^{(i)}, R_z^{(i)}] = \texttt{qr}(J_z^{(i)})$
    $[t^{(i)}, T^{(i)}] = [Q_z^{(i)}]^T [r, J_y]$
    $J_s^{(i)} = J_y^{(i)} - Q_z^{(i)} T^{(i)}$
end for
$[Q_s, R_s] = \texttt{qr}(J_s)$
$\Delta y = -R_s^{-1} Q_s^T r$
for $i = 1, \ldots, n$ do
    $\Delta z^{(i)} = -[R_z^{(i)}]^{-1}(t^{(i)} + T^{(i)}\Delta y)$
end for
```

2.4.2 Mixed CG/Direct method for systems with one very large block.

In some separable inverse problems, the number of linear variables z is too large for direct solution by Cholesky or QR factorization. This is particularly true in image and volume reconstruction problems: if each pixel of a 256×256 pixel image is considered a free variable, which is very modest by imaging system standards, the relevant Jacobians and Hessians will be 65536×63356 and usually impossible to factorize or even store in memory. In this case conjugate gradients (CG) or other iterative linear algebra methods must be employed to solve the Newton-type systems $B\Delta x = -g$. These methods only need functions that compute matrix-vector products with B, which may be much less memory consuming if B has special structure. Unfortunately the matrix B is often ill-conditioned, which can lead to slow convergence of CG. In some cases, B_{zz} is well conditioned, but the additional blocks involving the nonlinear variables y result in a poorly conditioned B. A method that uses iterative linear algebra only on the subblock B_{zz} has the potential to be more efficient.

Such a method may be derived by solving $B\Delta x = -g$ in the block decomposed form (2.7). We first solve (2.7a) by forming the small matrix $B_s = B_{yy} - B_{yz}B_{zz}^{-1}B_{zy}$ column-by-column. We solve for Δy by Cholesky factorizing this matrix, then solve (2.7b) by CG to obtain Δz, as summarized in Algorithm 2.8. To understand when Alg. 2.8 may be more efficient than full CG, we roughly estimate and compare the costs of each algorithm. Let t be the total floating point operations (flops) required to compute a matrix-vector product with B. We split t into $t = t_y + t_z$, where t_z is the cost of a matrix-vector product with B_{zz}, and t_y is the cost of computing products with all three remaining blocks B_{yy}, B_{yz}, and B_{zy}. Then solving $B\Delta x = -g$ requires $T_{cg} = k(t_y + t_z)$ flops, where k is the number of iterations required to achieve some suitable accuracy.

Algorithm 2.8 Mixed CG/Direct solution of $B\Delta x = -g$.

```
Require: Functions that compute matrix-vector products with $B_{yy}$, $B_{yz}$, $B_{zy}$, $B_{zz}$.
    Inverse matrix-vector products $B_{zz}^{-1}w$ are computed by conjugate gradients.
for $i = 1, \dots, N_y$ do
    $(B_s)_{:,i} = B_s e_i$
end for
Calculate a Cholesky factorization $R^T R = B_s$
$g_r = g_y - B_{yz}B_{zz}^{-1}g_z$
$\Delta y = R^{-1}R^{-T}g_r$
$\Delta z = -B_{zz}^{-1}(g_z - B_{zy}\Delta y)$
```

In Alg. 2.8, we assume that computing B_s is the dominant cost and the other computations are negligible, which is reasonable if N_y is significantly greater than 1. If k_z is the number of CG iterations required to solve $B_{zz}u = w$ to suitable accuracy, then the cost of computing each column of B_s is $t_y + k_z t_z$, yielding a total cost of $T_{mix} = N_y(t_y + k_z t_z)$ for all N_y columns. By setting $T_{mix} \leq T_{cg}$, we see that Alg. 2.8 will outperform full CG when the iterations k required by full CG exceeds a certain threshold:

$$k \gtrsim N_y \frac{t_y + k_z t_z}{t_y + t_z}. \tag{2.37}$$

The right-hand side is smallest when t_y is much larger than t_z, k_z, and N_y is relatively small; this corresponds to the case where B_{zz} is relatively well conditioned, products with B_{zz} are cheap, and there are not too many parameters in y. If $t_y \gg t_z$, then the threshold becomes $k \gtrsim N_y$. This is the minimum number of iterations we would expect from full CG if the eigenvalues of B_{yy} are isolated, so Alg. 2.8 should perform at least as well as full CG in this limit. However, if the spectrum of B_{zz} and the other blocks combines unfavorably, the required iterations k could be much larger, in which case Alg. 2.8 should be more efficient.

Even when (2.37) does not hold, Alg. 2.8 may still be desirable for other reasons. For example, if B is much more ill-conditioned than B_{zz}, round-off error will be less severe in Alg. 2.8 than in full CG because direct linear algebra is less vulnerable to bad conditioning. Also, Alg. 2.8 is highly parallelizable because each column of B_s can be computed completely independently of the others, while full CG is an inherently sequential algorithm.

2.5 Numerical experiments

In this section we show how semi-reduced methods can help us solve practical scientific problems faster and more robustly. To this end, we consider two model inverse problems relevant to scientific applications. In these problems, the use of Poissonian likelihoods and/or bound constraints greatly increases solution accuracy, so the unconstrained least squares is not preferable and reduced update methods are not appropriate. They are also well-suited for the linear algebra methods derived in §2.4. The first problem is an exponential sum fitting problem involving multiple measurement vectors, and the second is a semiblind deconvolution problem from solar astronomy. We also solve a third problem, which is a toy model of the second problem. Its purpose is to show when trial point adjustment can be useful, since (as discussed below) we did not find it particularly useful in the first two problems.

For each of the three problems, we selected an appropriate semi-reduced method and compared it to a standard full update method. In the first two problems, the full update method was the projected Newton-type method Alg. 2.5 with no block Gaussian elimination and no block trial point adjustment. This approach was compared with two alternatives: Alg. 2.5 with elimination off and adjustment on, and Alg. 2.5 with elimination on and adjustment off. (Elimination and adjustment act independently, so testing a fourth condition with both techniques switched on yields little additional information.) Block Gaussian elimination was performed using one of the methods derived in §2.4, while block trial point adjustments were obtained by performing a few iterations of Alg. 2.5 to approximately solve $\min_{z \in \mathcal{Z}} F(y^k, z)$, starting from the current iterate z^k. The parameters were set to $\delta = 10^{-4}$, $\alpha = 0.2$, $\lambda_{min} = 10^{-20}$, $\lambda_{max} = 10^{20}$, $\epsilon_0 = 2.2 \cdot 10^{-14}$, $\rho_{good} = 0.7$, $\rho_{bad} = 0.01$, $\tau = \max(2.2 \cdot 10^{-15}, \|\mathcal{P}(x^0 - \nabla F(x^0)) - x^0\|/10^8)$ where x^0 is the initial point. All of our experiments were performed in MATLAB R2011a on a MacBook Pro with 2.4 GHz Intel Core 2 Duo processor.

Our first finding was that trial point adjustment did not help us to solve the first two problems faster. Adjustment sometimes reduced backtracking and the total number of outer iterations needed, but not consistently or dramatically enough to outweigh the cost of solving inner adjustment subproblems at every iteration. As a result, total function evaluations and total runtime generally increased significantly when adjustment was used. For example, over 20 randomly sampled instances of the problem in §2.5.1, we compared a stringent inner solver ($k_{max} = 100$, $\tau = \|\mathcal{P}(z^k - \nabla_z F(y^k, z^k)) - z^k\|/10^8$) to no inner solver at all; in the former case the total function evaluations to solve the problem ranged from $200 - 600$, while for no inner iterations the range was $60 - 100$. We tried various intermediates between these two extremes—intermediate values of τ, lower values of k_{max}, stopping early if the Armijo condition was satisfied before the inner iteration limit—but we always found that it was most efficient to simply set $k_{max} = 0$, meaning no trial point adjustment.

For this reason we do not report any further on the effects of trial point adjustment in the first two problems. Instead we focus on the effects of block Gaussian elimination in the first two problems, and consider adjustment's effects only in the third problem. Note that since high-precision inner optimizations generally cause inefficiency in the first two problems, extensions of variable elimination that require them (such as [32, 96]) would be vulnerable to inefficiency in these problems, even if they could handle nonquadratic objectives.

2.5.1 Exponential sum fitting

In exponential sum fitting problems, the expected value $\mu(t)$ of a physical quantity at time t is assumed to be the sum of c exponentially decaying components with decay rates y_j and nonnegative weights z_j:

$$\mu(t) = \sum_{j=1}^{c} z_j \exp(-y_j t). \tag{2.38}$$

In many cases the decay rates do not vary from experiment to experiment, but the weights z may vary [80]. Thus, if n experiments are performed, the expected decay in the k^{th} experiment is

$$\mu_k(t) = \sum_{j=1}^{c} Z_{jk} \exp(-y_j t), \quad k = 1, \ldots, n. \tag{2.39}$$

We assume that a set of m Poisson-distributed observations $B_{1k}, \ldots, B_{mk}$ of each $\mu_k(t)$ are made at $t = t_1, \ldots, t_m$:

$$B_{ik} \sim \text{Poisson}(\mu_k(t_i)), \quad \text{for} \quad \begin{matrix} i = 1, \ldots, m \\ k = 1, \ldots, n. \end{matrix} \tag{2.40}$$

If the columns of B and Z are stacked on top of each other to form vectors b and z, then the associated maximum likelihood problem is

$$\underset{y,z}{\text{minimize}} \quad L((I_n \otimes A(y))z) \quad \text{subject to} \quad z \geq 0, \tag{2.41}$$

where $A(y)_{ij} = \exp(-y_j t_i)$, $\otimes$ is the Kronecker product, I_n is the $n \times n$ identity matrix, and $L(\mu)$ is the Poisson negative log-likelihood.

Using this model we generated synthetic data which simulated the problem of determining several decay rates from a large collection of relatively low-count time series. Each time series was generated from $c = 4$ decaying components with rates $(y_1, y_2, y_3, y_4) = (1, 2, 3, 4)$ and $m = 1000$ uniformly spaced time samples from $t = 0$ to 5. The number of measurement vectors was $n = 100$, and the nonnegative weights were randomly generated according to $z_{jk} = 10 \exp(1.2 \mathcal{Z}_{jk})$, where the $\mathcal{Z}_{jk}$ were random numbers from the standard normal distribution. A typical curve generated by this model is shown in Fig. 2.2. While this simple model does not directly represent a real physical problem, it generates problems similar in mathematical form, scale, and difficulty to problems encountered in real data analysis [78, 80]. In particular, each component has a few measurement vectors in which it dominates, but no component is ever observed in complete isolation. The persistent mixture of components with similar rates and the low signal-to-noise ratio combine to make this problem formidable.

As we mentioned above, trial point adjustment was not useful in this problem, so here we compare Alg. 2.5 in two modes: a semi-reduced mode with block Gaussian elimination, and a full update mode without it. In both cases the Hessian model was computed using the Gauss-Newton method [19,108], and the resulting Gauss-Newton system was solved by QR. (Direct Cholesky factorization of the normal equations is not sufficiently accurate due to the notoriously poor conditioning of exponential fitting problems [45].) In the standard mode, the full normal equations were solved directly using MATLAB's built-in sparse QR routine, while in the block Gaussian elimination mode, we used a MATLAB implementation of Alg. 2.7. MATLAB sparse QR employs the state-of-the-art SuiteSparseQR package [33]. To obtain the best possible performance from SuiteSparseQR, matrix-vector products with the Q factor were performed implicitly, and a permutation was applied to switch the blocks J_y and J_z. (The permutation speeds up the algorithm by an order of magnitude, as it enables the underlying Householder triangularization method to preserve the matrix's sparsity pattern.) Note that Alg. 2.7 has a less efficient implementation than SuiteSparseQR because the loops in Alg. 2.7 run relatively inefficiently in MATLAB, while SuiteSparseQR is written in C++.

Our main finding was that block Gaussian elimination computed steps several times faster than sparse QR with no loss of accuracy. In a typical random instance of the problem described above, step computation by sparse QR factorization of the full Jacobian required 0.38 seconds (s), while Alg. 2.7 solved the system in 0.10 s, a roughly 4-fold improvement. Since most of the algorithm's time is spent in step computation, the minimum was found significantly faster using block Gaussian elimination: in this instance, the standard mode took 18 s, while using Alg. 2.7 took 6 s. The accuracies of the two modes were functionally indistinguishable, as the objective values $F(y^k, z^k)$ output in each mode were the same to at least 8 significant figures. From this we infer that the two algorithms do essentially the same mathematical operations, but the computer finishes the operations faster using Alg. 2.7.

The speed difference can be explained by two factors. First, Alg. 2.7 does not build the full J matrix, but factorizes of the n diagonal blocks of J_z just as they are needed. In contrast, the sparse QR algorithm must build all of J first, which takes $60 - 80\%$ of the CPU time required to actually solve the system. In Alg. 2.7 the blocks of J_z are built and factorized just-in-time, so there is no need to build a large sparse matrix. Second, Alg. 2.7 solves the overall system by solving a large number of small and very similar subsystems, which is more CPU and cache-friendly than operating on a large sparse matrix.

The formidable difficulty of this problem, and the need for a bound-constrained Poissonian solver, may be appreciated by comparing the accuracy of the Poissonian method to a popular alternative for Poissonian problems, the variance-weighted least

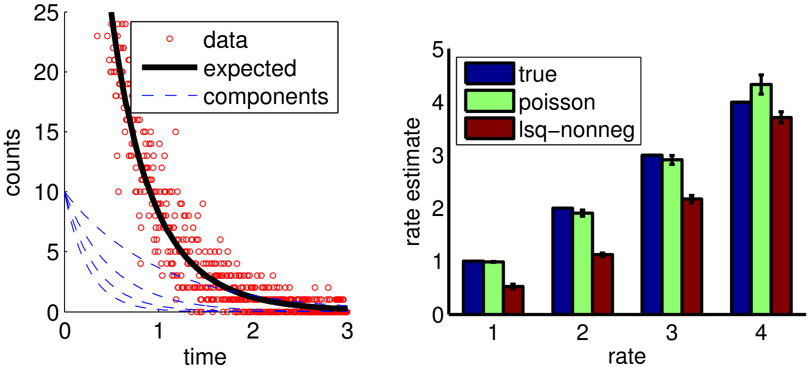

Figure 2.2: *Left:* Sample data from the sum-of-exponentials model. The four decaying components (blue dotted lines) have decay rates $y_j = j$ for $j = 1, 2, 3, 4$, and when summed together with weights z_j, these components create the expected intensity curve $\mu(t)$ (solid black line). The Poisson-distributed samples b_i of $\mu(t)$ (red dots) are taken at a spacing of $\Delta t = 0.005$. The low available counts suggest a Poisson likelihood should be used. *Right:* Comparison of fitted and true decay rates y_j for $j = 1, 2, 3, 4$ using variance-weighted nonnegative least squares and Poisson likelihood. The bar heights are the median values found by solving 100 random problem instances, and the error bars represent median absolute deviations.

squares method. In the variance-weighted least squares method one solves

$$\underset{y,z}{\text{minimize}} \quad \|W[(I_n \otimes A(y))z - b]\|_2^2 \quad \text{subject to} \quad z \geq 0, \qquad (2.42)$$

where W is a diagonal matrix with $W_{ii} = 1/\max(b_i^{1/2}, \epsilon)$, and $\epsilon = 1$ is a small constant used to avoid division by zero [69]. We generated 100 random instances of the exponential sum fitting problem described above, and solved each using the Poissonian approach (2.41) and the weighted least squares approach (2.42), in both cases using Alg. 2.5. The decay vector y resulting from each experiment was sorted to account for the problem's permutation ambiguity, resulting in 100 estimates of y_1, y_2, y_3, and y_4 from each method. We then calculated the median and median absolute deviation of the 100 estimates of each y_i from each method. (We used the median as a summary statistic because it is invariant to reparametrization of y and robust to the occasional failures of both methods.) The results are shown in Fig. 2.2, *right*, and it is clear that the Poissonian solver's decay rates are far more accurate.

2.5.2 Multiframe semiblind deconvolution

Image deconvolution is a linear inverse problem in which we have an image b degraded by convolution with a known point spread function (PSF) h, and we wish to undo the degradation to obtain the unknown clean image z. Assuming that each of these variables are 2D arrays supported on a square $\Omega \in \mathbb{Z}^2$, we can write the problem as

$$Az + \epsilon = b, \tag{2.43}$$

where $A : \mathbb{R}^\Omega \to \mathbb{R}^\Omega$ is the convolution operator: $Az = h * z$, and we assume periodic boundary conditions for simplicity. In multiframe blind deconvolution, there are several images and PSFs and the PSFs depend on unknown parameters, so that we have

$$A(y^{(k)})z^{(k)} + \epsilon^{(k)} = b^{(k)} \quad \text{for } k = 1, \ldots, n. \tag{2.44}$$

If we have a parametric model of the PSFs, the problem is called semiblind.

Here we consider a simplified, synthetic version of a real multiframe semiblind deconvolution problem from solar imaging, which is described in [94]. In this problem, a spaceborne telescope observing the Sun in the extreme ultraviolet wavelengths collects images which are are contaminated by stray light. The stray light effect is well-modeled by convolution with a single unknown parametric PSF. The telescope observes n images of the Moon transiting in front of the Sun, and while the Moon does not emit in the extreme ultraviolet (Fig. 2.3, *top middle*), stray light from the Sun spills into the lunar disk (*bottom middle*). Given the supports $M^{(k)} \subset \Omega$ of the lunar disks within each image, our task is to determine the PSF by solving

$$\underset{y,\{z^{(k)}\}}{\text{minimize}} \quad \sum_{k=1}^{n} \|A(y)z^{(k)} - b^{(k)}\|^2 \quad \text{subject to} \quad \begin{matrix} z^{(k)} \geq 0 \\ z^{(k)}_{M^{(k)}} = 0 \end{matrix} \quad \text{for } k = 1, \ldots, n. \tag{2.45}$$

The PSF is modeled using two components. The PSF core is modeled by a single pixel with unknown value $\alpha \in (1/2, 1]$, while the wings are modeled by a radially symmetric piecewise power law $p_{\boldsymbol{\beta}}(r)$ depending on unknown parameters $\boldsymbol{\beta}$:

$$h_y(v) = \alpha\delta_0(v) + (1-\alpha)p_{\boldsymbol{\beta}}(\|v\|_2), \quad \text{for } v \in \Omega, \tag{2.46}$$

where δ_0 is the Kronecker delta. To define the piecewise power law, we set $p_{\boldsymbol{\beta}}(0) = 0$, then for $r > 0$ we set logarithmically spaced breakpoints $(r_i)_{i=0}^S$ defining $S = 12$ subintervals, starting from $r_0 = 1$ and ending at $r_S = \frac{\sqrt{2}}{2}s$ where s is the sidelength of the square Ω. On each subinterval $[r_{i-1}, r_i)$, the formula is given by $p_{\boldsymbol{\beta}}(r) \propto r^{-\beta_i}$,

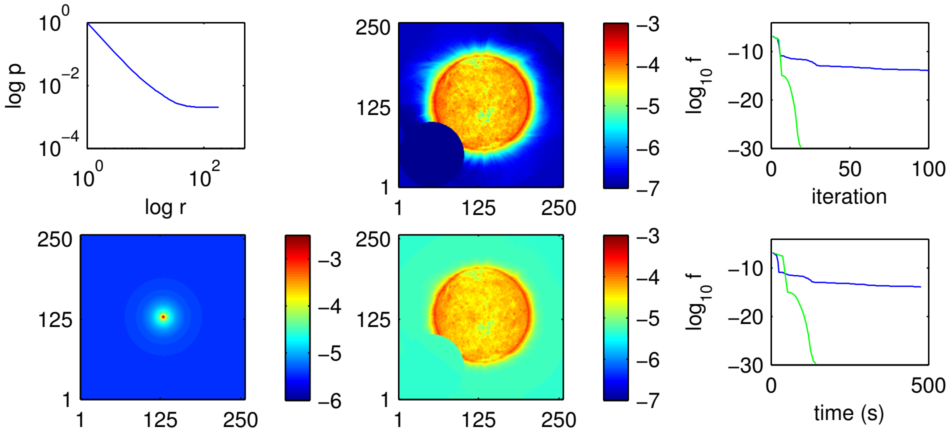

Figure 2.3: Overview of the solar semiblind deconvolution experiment. *Top left:* The ground truth PSF profile $p^{true}(r)$ in log-log scale, where it is piecewise linear. *Bottom left:* The ground truth PSF generated by the profile above. *Top middle:* one of the three clean lunar transit images, with lunar disk in the bottom left corner (logarithmic scale). *Bottom middle:* the observed image formed by convolving the top image with the PSF (logarithmic scale). *Right:* semilog plot of objective versus iteration *(top)* and CPU time *(bottom)* for the standard mode of Alg. 2.5 and the mode employing the mixed CG/Direct method.

where $\beta_i \geq 0$, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_S)$. The proportionality constants are determined by a continuity constraint between subintervals and the normalization constraint $\sum_v p_{\boldsymbol{\beta}}(\|v\|) = 1$. The free parameters of the PSF model h_y are then $y = (\alpha, \boldsymbol{\beta}) \in \mathbb{R}^{N_y}$, where $N_y = 1 + S = 13$. The true profile $p_{\boldsymbol{\beta}}(r)$ was generated using $\boldsymbol{\beta}$ values similar to those in [94], and is shown in log-log scale in (Fig. 2.3, *top left*), with the resulting PSF directly below.

We used data from the STEREO-EUVI satellite to generate $n = 3$ synthetic lunar transit images of size 256×256. To simulate the Moon's transit, a disk of pixels was set to zero in each image. These images were convolved with the ground truth PSF to create the blurry observations, and no noise was added for simplicity. (Noise is not a very important issue in this problem because the real data have very little noise at this resolution, deconvolution with the PSF is well-conditioned for $\alpha > 1/2$, and the expected range of α is well above this.)

As before, Alg. 2.5 was run in two modes: a semi-reduced mode employing Gaussian elimination, and a standard one without. In the standard mode of Alg. 2.5, the search direction was calculated by CG on the full system $B\Delta x = -g$. Preliminary experiments revealed that the full CG algorithm was very slow. However the situation improved substantially when we used a scalar preconditioner cI on the y block, where

$c = 10^5$ was found to work well. All CG iterations were stopped at a relative residual tolerance of 10^{-6} and maximum iteration ceiling 40, as these values worked relatively well for the full CG algorithm. In the block Gaussian elimination mode, the search direction was calculated using the mixed CG/Direct algorithm, Alg. 2.8. The mixed CG/Direct algorithm required no special tuning or preconditioners.

Our main finding was that the block Gaussian elimination mode using Alg. 2.8 converged quite quickly and robustly, while the standard mode experienced a long period of sluggish convergence after an initially fast descent (Fig. 2.3, *right*). The average CPU time per step was about the same in each of the two modes, so we can attribute the block Gaussian elimination mode's superior performance to better search directions, which enabled convergence in far fewer iterations than the full CG mode.

The better search directions of the block Gaussian elimination mode can be explained by considering the unusual spectrum of the Gauss-Newton Hessian. It has two very different components: a large cluster of $\approx N_z$ near-unity eigenvalues due to the very well-conditioned B_{zz} block, and a sprinkling of $\approx N_y$ eigenvalues contributed by the other three blocks. The latter are less tame: they can easily spread over 15 orders of magnitude and move unpredictably as the iterations progress.

Naively, we might expect full CG to make short work of such a system. We simply apply a scalar preconditioner to the badly behaved blocks involving Δy, pushing the N_y scattered eigenvalues to lie above the N_z cluster. Then the spectral theory of CG predicts convergence in $N_y + k_z$ iterations, where k_z is the number of iterations required to make the CG spectral polynomial nearly zero on the N_z cluster [101]. We expect k_z to be small because the N_z cluster is very tightly centered around unity.

In practice, however, it is difficult to know in advance where the mobile eigenvalues will be, and their enormous spread raises issues of rounding error. Thus it is difficult to get good solutions out of full CG, and the search directions suffer, causing sluggish convergence. In constrast, the mixed CG/Direct algorithm applies CG to the well-conditioned B_{zz} block alone, and deals with the other blocks by direct linear algebra. Since direct linear algebra is much less susceptible than CG to ill-conditioning and rounding error, the result is high-quality search directions and quick convergence.

2.5.3 A model semiblind deconvolution problem for block trial point adjustment

Given the failure of trial point adjustment to speed up the solution of the previous two problems, the reader may wonder if it has any application beyond its theoretical role in the connection between full and reduced update methods. The literature suggests that adjustment certainly can increase convergence rate and robustness [44, 67, 84, 90, 97]. However the speed gains relative to standard methods are highly variable: adjusted methods are slower in our experiments, a factor of 2 or 3 times faster in certain image processing problems, and multiple orders of magnitude faster in some difficult curve fitting problems. Clearly adjustments must be adapted to the problem at hand, but it is difficult to predict when it will be useful. Here we present a toy semiblind deconvolution problem similar to the one solved in the previous section, and show that trial point adjustment is valuable for solving this problem in the most difficult cases.

As in the solar problem, our toy problem involves semiblind deconvolution of an extended, uniformly bright object which has been convolved with a long-range kernel. The true image u^t and kernel h^t are both 1-D signals of length m supported on $\{-j, \ldots, j\}$, where $m = 2j+1$. They come from single-parameter signal families given by $h_y(p) = y\delta_0(i) + (1-y)\frac{1}{m}\mathbf{1}(i)$ and $u_z(i) = z \cdot \mathbf{1}_S(i)$, where $\mathbf{1}_S(i)$ is the indicator for the set $S = \{-\ell, \ldots, \ell\}$ of size $s = 2\ell + 1$, $\mathbf{1}(i)$ is the constant ones function. Letting (y^t, z^t) denote the unknown true parameter values, the problem is to determine (y^t, z^t) from the blurry observation $f = h^t * u^t$, where periodic convolution and no noise is assumed. The values of y^t and z^t can be found by minimizing the difference between $h_y * u_z$ and f with respect to some loss function, which we choose as the Huber loss

$$\ell(\mu) = \begin{cases} \frac{1}{2}\mu^2 & |\mu| \leq \theta \\ \theta(|\mu| - \frac{\theta}{2}) & |\mu| > \theta \end{cases} \tag{2.47}$$

with threshold $\theta = 0.3$. We choose this loss function simply because it is a common nonquadratic loss and the optimization phenomenon of interest occurs when it is used. Noting that physically we must have $0 \leq y \leq 1$ and $z \geq 0$, we obtain the optimization problem

$$\min_{0 \leq y \leq 1, z \geq 0} \left\{ F(y,z) \triangleq \sum_i \ell((h_y * u_z - f)_i) \right\}. \tag{2.48}$$

A simple formula for $F(y,z)$ can be found by observing that both the prediction $h_y * u_z$ and f take only two values. Letting $\rho = s/m$ be the ratio of the object support to signal size, $q_1(y,z) = yz + (1-yz)\rho$ the predicted value of the blurry image on the support, $q_2(y,z) = (1-y)z\rho$ the predicted value off the support, and $q_i^t = q_i(y^t, z^t)$ for $i = 1, 2$, we have

$$F(y,z) = \frac{m}{2}\left(\rho\ell(q_1(y,z) - q_1^t) + (1-\rho)\ell(q_2(y,z) - q_2^t)\right). \tag{2.49}$$

The $m/2$ scale factor does not affect the location of the minimum nor the path of any of the optimization algorithms we consider here. Therefore, for our purposes, the parameter ρ is effectively the only free parameter in the problem family. We use the values $\rho = 10^{-2}, 10^{-6}$ to create two objectives whose graphs are depicted in Fig. 2.4, *far left.* As $\rho \to 0$, the term $\rho\ell(q_1(y,z) - q_1^t)$ vanishes, the $(1-\rho)\ell(q_2(y,z) - q_2^t)$ becomes dominant, and the objective landscape becomes a narrow, hyperbolic trench.

We solved this problem at both values of ρ, and for each value we used Alg. 2.5 in a full update mode (without trial point adjustment) and a semi-reduced mode (with trial point adjustment). In the latter case, the block trial point adjustment used was a single iteration of Alg. 2.5 to minimize $F(y,z)$ in z with y fixed. Algorithm parameters were chosen as in the previous section.

The paths taken by the full and semi-reduced methods are shown in Fig. 2.4, *center left and right.* We observe that the methods take nearly identical paths when $\rho = 10^{-2}$, but when $\rho = 10^{-6}$ the full update method is forced to take very small steps. At *far right*, the distance to the minimum, $\|(y^k, z^k) - (y^t, z^t)\|_2$, is plotted versus iteration k for each method. The superior convergence rate of the semi-reduced method is clear when $\rho = 10^{-6}$.

The behavior of each algorithm can be understood by considering the geometry of the steps it takes. The full update method takes steps along straight lines. Straight lines cannot follow a curved trench for long, so there is an upper bound on the size of an admissible step. As $\rho \to 0$, the trench tightens and the admissible steps become very small, so that progress is very slow. The semi-reduced method takes a 'dogleg' step as illustrated in Fig. 2.1, which enables it to stay in the valley. (Here dogleg refers to the appearance of the step alone, with no relation to the dogleg step used in trust region methods.)

To avoid the small admissible step issue that stymies the full method, it is critical that adjustment be done *before the trial point is evaluated.* This is the key feature distinguishing semi-reduced methods from other methods, such as simple alternation between a full update and a partial update. Other strategies, such as nonmonotone line search [114] and greedy two-step methods [30], have a similar step structure and could also work on this problem; however it is unclear if they can match the semi-reduced method's complete insensitivity to the value of ρ.

It is important to note that the phenomenon we have described here does not occur for all loss functions $\ell(\mu)$. For example, we found that if the Poisson log-likelihood is used, the objective landscape does not have such a tight curved valley, the full update method solves the problem quite efficiently, and the semi-reduced method's

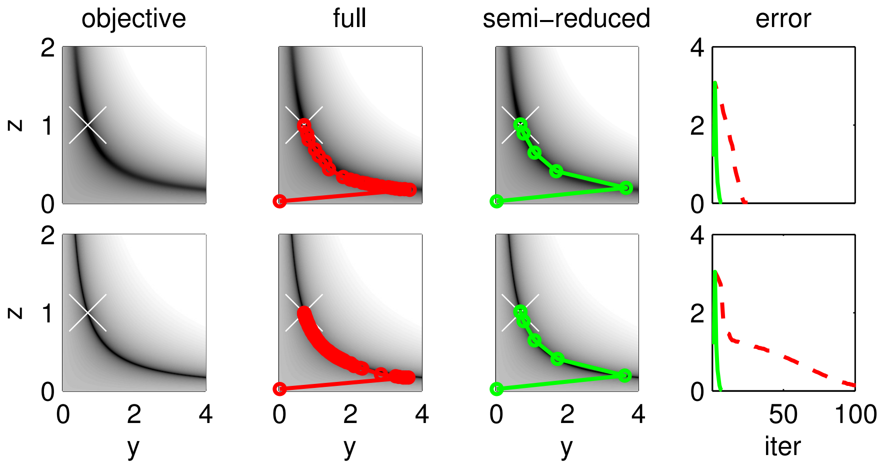

Figure 2.4: Comparison of full and semi-reduced methods on a toy blind deconvolution problem. *Row 1:* Plots of $F(y, z)$ for $\rho = 10^{-2}$ (top) and 10^{-6} (bottom), logarithmic greyscale. The white crosses mark the minimum at $(y^t, z^t) = (0.7, 1)$, where $F = 0$. *Center left and right:* The iterates of the full and semi-reduced methods for each ρ value, starting from $(y^0, z^0) = (0.02, 0.02)$. *Far right:* semilogarithmic plot of the error $\|(y^k, z^k) - (y^t, z^t)\|_2$ versus iteration k for the full method (dashed line) and semi-reduced method (solid line).

inner iterations expend effort without benefit. Curved valleys are thus an occasional problem with potentially severe efficiency consequences. The semi-reduced framework seems appropriate for dealing with such a problem, since one has the option to perform inner descent iterations only when necessary.

2.6 Conclusion

Reduced update optimization methods, which are based on variable elimination, have been found to be particularly fast and robust in certain difficult separable inverse problems. Unfortunately, using them in problems beyond unconstrained least squares presents serious theoretical and practical difficulties, in particular the need for expensive optimal trial point adjustments and complex derivatives of a reduced functional. We have described a new class of *semi-reduced* methods which interpolate between full and reduced methods. Semi-reduced methods share the desirable characteristics of reduced methods while being flexible enough to avoid their downsides. A key advantage of the semi-reduced framework is the flexibility to use adjustments where they are useful and avoid them where they are not, all within a single convergent method.

We began by reinterpreting reduced methods as full update methods that have been modified and simplified. We showed that if *block Gaussian elimination* and an optimal *block trial point adjustment* are used within a full update method, the adjustment's optimality renders certain computations unnecessary. Removing these unnecessary computations yields a simplified method that turns out to be equivalent to a reduced method. To confirm that this reinterpretation of reduced update methods is correct and generally applicable, we derived the well-known reduced update Newton and variable projection methods using our modification and simplification process. We defined semi-reduced methods by omitting the final simplifications, which frees us from the need to perform expensive optimal block trial point adjustments. We then incorporated block Gaussian elimination and trial point adjustment into an algorithm for general bound constrained problems, and showed that its convergence follows almost immediately from the convergence theorem for the original method. Finally, we showed that many of the structure-exploiting properties of variable elimination can be obtained by using appropriate block Gaussian elimination algorithms.

Block Gaussian elimination is suited for problems where the Hessian model's B_{zz} subblock is block diagonal, circulant, banded, or has other exploitable structure. We described two situations where we expected block Gaussian elimination to outperform a standard all-at-once method, and these expectations were borne out in numerical experiments on realistic problems derived from the scientific inverse problem literature. It is notable that both of the methods we presented involve the solution of many independent subproblems and are thus ideal candidates for parallelization.

Block trial point adjustment is appropriate when we expect the graph of $F(y, z)$ to contain a narrow, curved valley. Trial points from full update methods tend to leave the valley and thus will be rejected unless a trial point adjustment is used to return to the valley. In our first two numerical experiments trial point adjustments turned out to be computationally wasteful, so it was critical that we had the flexibility to perform suboptimal adjustments or even none at all (which turned out to be the best option). In our third experiment we presented a reasonable toy inverse problem where the curved valley effect was significant enough to warrant trial point adjustment, but the parameter values where this occurred were somewhat extreme. Since the curved valley effect is important in some real problems [30, 67], a better understanding of precisely when it occurs would be useful.

CHAPTER 3

Correcting Camera Shake by Incremental Sparse Approximation

3.1 Introduction

A fundamental problem in image processing for handheld cameras is the correction of blur caused by movement of the camera during the exposure. This chapter is concerned with a new method for blind deblurring of images corrupted by camera shake.

This problem is very different from the solar imaging problem discussed in the next chapter, and must be treated with very different signal models and optimization methods. In the solar problem, motion of the imaging instrument is not an issue and blur arises from optical imperfections that handheld cameras are not subject to. Solar blur kernels have a support comparable to the full image, which is far too large for fully nonparametric modeling. However, they are sufficiently regular and well-understood to be reasonably approximated by parametric modeling. Information about the kernel comes from special events where the true image is partially known. The resulting optimization problems are smooth, but their ill-conditioned and highly non-diagonal Hessians cause alternating and even full update methods to converge slowly, making reduced and semi-reduced methods preferable.

Camera shake kernels have a relatively small but highly irregular support and are generally modeled fully nonparametrically, leading to a much larger set of kernel parameters than in the solar problem. Information about the kernels is extracted by assuming that the image contains a collection of objects separated by a sparse set of sharp edges. The sparse edge assumption is typically modeled mathematically

A version of this chapter has been accepted for publication in ICIP 2013 with co-authors Anna C. Gilbert and Alfred O. Hero III.

by a sparsity-promoting ℓ_p prior on the edges, leading to nonsmooth optimization problems. The nonsmoothness, nonconvexity, and large number of kernel parameters make this problem difficult to address by any optimization method other than alternating update.

A camera generally does not record how it was moved during an exposure, so the correction of camera shake is a *blind deconvolution* problem: we are given a blurry image y and must determine an estimate x of the unknown sharp image x^{true} without knowledge of the blur kernel. In the simplest model of blur, y is formed by convolving x^{true} with a single blur kernel k^{true} and adding noise n:

$$y = k^{\text{true}} * x^{\text{true}} + n. \tag{3.1}$$

This convolution model assumes that k^{true} does not change with position, an assumption which is frequently violated due to slight camera rotations and out-of-plane effects [71]. Still, the uniform model works surprisingly well and methods for it can be extended to handle nonuniform blur [25, 109].

The camera shake blind deconvolution problem is highly underdetermined and additional assumptions must be made to obtain a solution. These assumptions are often imposed most conveniently by moving the problem into a filter space. We define filters $\{f_\gamma\}_{\gamma=1}^{L}$ and set $y_\gamma = f_\gamma * y$ and $x_\gamma^{\text{true}} = f_\gamma * x^{\text{true}}$, so that

$$y_\gamma = k^{\text{true}} * x_\gamma^{\text{true}} + n_\gamma \tag{3.2}$$

for $\gamma \in [L] = \{1, \ldots, L\}$. Defining $\mathbf{x}^{\text{true}} = \{x_\gamma^{\text{true}}\}_{\gamma=1}^{L}$, $\mathbf{y} = \{y_\gamma\}_{\gamma=1}^{L}$, and $(k * \mathbf{x}^{\text{true}})_\gamma = k * x_\gamma^{\text{true}}$, we can write the filter space problem compactly as

$$\mathbf{y} = k^{\text{true}} * \mathbf{x}^{\text{true}} + \mathbf{n}. \tag{3.3}$$

The simplest nontrivial filter space is gradient space, where $L = 2$ and $f_1 = [1, -1]$, $f_2 = [1, -1]^T$, but there are many other possibilities. Determining x from a filter space representation $\mathbf{x}$ often does not work well, so typically one obtains an estimate k of k^{true} and deconvolves y with k to get x [71].

Bayesian inference is a convenient framework for imposing prior assumptions to regularize blind deconvolution [21]. By assuming some distribution of $\mathbf{n}$ we obtain a likelihood function $p(\mathbf{y} \mid k * \mathbf{x})$ which gives the probability that the data $\mathbf{y}$ arose from a given pair $(k, \mathbf{x})$. We then choose priors $p(k)$ and $p(\mathbf{x})$ and compute the posterior distribution

$$p(k, \mathbf{x} \,|\, \mathbf{y}) \propto p(\mathbf{y} \,|\, k * \mathbf{x}) p(\mathbf{x}) p(k). \tag{3.4}$$

Estimates of $\mathbf{x}$ and k may be obtained by summary statistics on $p(k, \mathbf{x} \,|\, \mathbf{y})$. We call the mode of $p(k, \mathbf{x} \,|\, \mathbf{y})$ the joint maximum *a posteriori* (MAP) estimator, while the mode of the marginal $p(k \,|\, \mathbf{y}) = \int p(k, \mathbf{x} \,|\, \mathbf{y}) d\mathbf{x}$ is the kernel MAP estimator. Most blind deconvolution methods are nominally MAP estimators but do not actually find a global minimizer, as this is typically intractable and may even be counterproductive. We refer to any method organized around optimizing a posterior as a MAP method, while methods that actually find a global minimum will be called *ideal* MAP methods. Joint MAP methods typically attempt to minimize the cost function $F(k, \mathbf{x}) = -\log p(k, \mathbf{x} \,|\, \mathbf{y})$, which may be written (up to an irrelevant additive constant) as the sum of a data misfit and two regularization terms,

$$F(k, \mathbf{x}) = L(k * \mathbf{x}) + R_{\mathbf{x}}(\mathbf{x}) + R_k(k), \tag{3.5}$$

where each of these functions may take the value $+\infty$ to represent a hard constraint. Kernel MAP estimation is more difficult as it involves a high-dimensional marginalization, and it is typically approximated by variational Bayes or MCMC sampling [13].

Joint MAP estimation is the oldest, simplest, and most versatile approach to blind deconvolution [6, 23, 112], but initial joint MAP efforts on the camera shake problem met with failure [38], even when ℓ_p regularizers for $p < 1$ were used. In [71], Levin *et al* showed that the ℓ_p regularizer generally prefers blurry images to sharp ones: $\|\mathbf{y}\|_p^p < \|\mathbf{x}^{\text{true}}\|_p^p$, so that ideal joint MAP typically gives the trivial *no-blur* solution $(k, x) = (\delta_0, y)$, where δ_0 is the Kronecker delta kernel. The non-ideal joint MAP methods [24,93] somewhat compensate for the defects of ideal joint MAP by dynamic edge prediction and likelihood weighting, but benchmarking in [71, 72] showed that these heuristics sometimes fail.

In [38] Fergus *et al* developed a kernel MAP method with a sparse edge prior which was very effective for correcting camera shake. In [71] it was noted that marginalization over $\mathbf{x}$ seems to immunize ideal kernel MAP against the blur-favoring prior problem. More refined kernel MAP methods were recently reported in [72] and [7], and to our knowledge these two methods are the top performers on the benchmark 32 image test set from [71]. While these efforts have made kernel MAP much more tractable, it remains harder to understand and generalize than joint MAP, so it would be useful to find a joint MAP method that is competitive with kernel MAP on the camera shake problem.

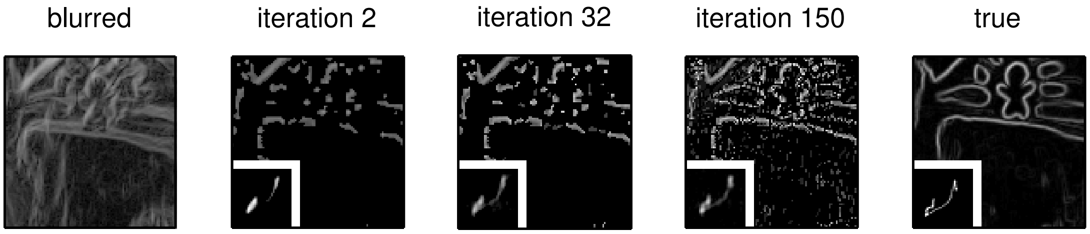

Figure 3.1: Kernel estimation on an image from the test set of [71]; a small patch has been selected and rescaled for clarity. *Left:* Blurry edge map $|\mathbf{y}|$. *Center left to center right:* Evolution of the kernel k (inset) and edge map magnitude $|\mathbf{x}|$ in the final full-resolution stage. As τ increases in (3.6), the edge map becomes less sparse and the kernel is refined. *Right:* k^{true} and $|\mathbf{x}^{\text{true}}|$.

In [66], Krishnan *et al* addressed the blur-favoring prior problem in joint MAP by changing the prior, proposing the scale-invariant ℓ_1/ℓ_2 ratio as a 'normalized' sparse edge penalty. The ℓ_2 normalization compensates for the way that blur reduces total ℓ_1 edge mass, causing the ℓ_1/ℓ_2 penalty to prefer sharp images and eliminating the need for additional heuristics. While their algorithm does not quite match the performance of [38] on the benchmark test set from [71], it comes fairly close while being significantly simpler, faster, and in some cases more robust. Other promising joint MAP methods include [20, 107, 110], but we are not aware of public code with full benchmark results for these methods.

3.1.1 Our approach

We propose a new approach to joint MAP blind deconvolution in which the kernel is estimated from a sparse approximation $\mathbf{x}$ of the sharp gradient map $\mathbf{x}^{\text{true}}$. Initially we constrain $\mathbf{x}$ to be very sparse, so it contains only the few strongest edges in the image, and we determine k such that $k * \mathbf{x} \approx \mathbf{y}$. Because $\mathbf{x}$ is so much sparser than $\mathbf{y}$, the solution $k = \delta_0$ is very unlikely. But generally this initial k overestimates k^{true}, so we refine k by letting weaker edges into $\mathbf{x}$.

To present this approach formally, we set $f_1 = [1, -1], f_2 = [1, -1]^T$, so that $\mathbf{x}(p) = (x_1(p), x_2(p))$ is the discrete image gradient vector at each pixel p. We set $L(k, \mathbf{x}) = \frac{1}{2}\|k * \mathbf{x} - \mathbf{y}\|_2^2$ and impose the usual positivity and unit sum constraints on k. We measure gradient sparsity using the $\ell_{2,0}$ norm: $|\mathbf{x}(p)|$ is the ℓ_2 length of $\mathbf{x}(p)$ and $\|\mathbf{x}\|_{2,0} = \||\mathbf{x}|\|_0$ the number of nonzero gradient vectors. The joint MAP optimization problem is then

$$\begin{array}{ll} \underset{k,\mathbf{x}}{\text{minimize}} & \frac{1}{2}\|k * \mathbf{x} - \mathbf{y}\|_2^2 \\ \text{subject to} & k \geq 0,\ 1^T k = 1,\ \|\mathbf{x}\|_{2,0} \leq \tau, \end{array} \tag{3.6}$$

where the expression $a^T b$ denotes the dot product of the arrays a and b when considered as vectors, and the 1 in $1^T k$ is an all-ones array.

We solve this problem with an iterative optimizer described in §3.2, and slowly increase τ as the iterations proceed. To initialize τ we use the ℓ_1/ℓ_2 ratio, a robust lower bound on a signal's sparsity [73]. The sharp $\mathbf{x}$ should be significantly sparser than $\mathbf{y}$, so initially we set $\tau = \beta_0 \tau_{\mathbf{y}}$, where $\tau_{\mathbf{y}} = \||\mathbf{y}|\|_1 / \||\mathbf{y}|\|_2$ and $\beta_0 < 1$ is a small constant. After an initial burn-in period of I_b iterations we multiply τ by a constant growth factor $\gamma > 1$, an action we repeat every I_s iterations thereafter.

We use a standard multiscale seeding technique to accelerate the kernel estimation step [38,66]. We begin by solving (3.6) with a heavily downsampled $\mathbf{y}$, giving a cheap, low-resolution k and $\mathbf{x}$. We then upsample these and use them as an initial guess to solve (3.6) with a higher resolution $\mathbf{y}$, repeating the upsample-and-seed cycle until we reach the full resolution $\mathbf{y}$. At each scale we use the same τ increase schedule. After kernel estimation we use non-blind deconvolution of y with k to get the sharp image x.

The easiest way to understand how our kernel estimation works is to watch k and $\mathbf{x}$ evolve as the iterations progress. In Fig. 3.1, the state of k and $\mathbf{x}$ is shown at iterations $2, 32$, and 150 of the final full-resolution scale, with k^{true} and $\mathbf{x}^{\text{true}}$ at far right. Initially $\mathbf{x}$ is quite sparse, so k cannot be a trivial kernel because the parts of $\mathbf{y}$ not in $\mathbf{x}$ must be attributed to blur. But this initial approximation is crude, so as τ increases with iteration, $\mathbf{x}$ is allowed to have more and more nonzeros so that k can be refined.

3.1.2 Novelty and relations with existing methods

Direct ℓ_0 optimization is well-established in the compressed sensing community [14, 42] but we are not aware of any effective ℓ_0 approaches to blind deconvolution. In [66] the ℓ_1/ℓ_2 ratio was deliberately chosen over ℓ_0 because while both have the desired scale invariance, the graph of ℓ_1/ℓ_2 is smoother and looks more 'optimizable' than ℓ_0. We contend that ℓ_0 may be difficult to use as a cost function, but very effective as a constraint. Gradient and kernel thresholding are commonly used [24,93] and these can be interpreted as ℓ_0 projections, but they are typically used as auxiliary heuristics, not as the central modeling idea. Our technique of slowly increasing the sparsity constraint τ is reminiscent of matching pursuit algorithms for sparse approximation [81, 102]. It is also related to the likelihood reweighting technique of [93], which may be seen by considering the Lagrangian of (3.6). However, our initialization strategy requires that we use the constrained formulation rather than the Lagrangian.

3.2 Alternating projected gradient method

To solve problem (3.6) at a given scale, we use a standard alternating descent strategy: starting from some initial k and $\mathbf{x}$, we reduce the cost function by updating $\mathbf{x}$ with k fixed, then k with $\mathbf{x}$ fixed, cycling until a stopping criterion is met. Each cycle, or outer iteration, consists of $I_{\mathbf{x}}$ inner iterations updating $\mathbf{x}$ and I_k inner iterations updating k. All updates are computed with a projected gradient method; given a smooth function $h(u)$ and a constraint set $\mathcal{U}$, projected gradient methods seek a solution of $\min_{u \in \mathcal{U}} h(u)$ by updates of the form $u \leftarrow \mathcal{P}_{\mathcal{U}}(u - \alpha_u g_u)$, where $g_u = \nabla h(u)$, α_u is a step size, and $\mathcal{P}_{\mathcal{U}}(w) = \operatorname{argmin}_{u \in \mathcal{S}} \|u - w\|_2^2$ is the Euclidean projection of w onto $\mathcal{U}$. Convergence of alternating descent and projected gradient methods to stationary points is proven in [4] under mild conditions.

We now describe how we compute the projected gradient iterations for the inner subproblems $\min_{k \in \mathcal{K}} L(k, \mathbf{x})$ and $\min_{\mathbf{x} \in \mathcal{X}} L(k, \mathbf{x})$, where $L(k, \mathbf{x}) = \frac{1}{2}\|k * \mathbf{x} - \mathbf{y}\|_2^2$, $\mathcal{K} = \{k \mid k \geq 0, 1^T k = 1\}$, and $\mathcal{X} = \{\mathbf{x} \mid \|\mathbf{x}\|_{2,0} \leq \tau\}$. Letting $\mathbf{r} = k * \mathbf{x} - \mathbf{y}$ denote the residual, we have $\nabla_k L = \sum_\gamma \bar{\mathbf{x}}_\gamma * \mathbf{r}_\gamma$ and $\nabla_{\mathbf{x}} L = \bar{k} * \mathbf{r}$, where the bar denotes 180° rotation about the origin. Assuming the nonzero elements of $|\mathbf{x}|$ are distinct, the projection $\mathcal{P}_{\mathcal{X}}(\mathbf{x})$ is the top-τ vector thresholding

$$P_{\mathcal{X}}(\mathbf{x})(i) = \mathbf{x}(i) \cdot \mathbf{1}\left(|\mathbf{x}(i)| \geq \theta(|\mathbf{x}|, \tau)\right), \tag{3.7}$$

where $\mathbf{1}(A)$ is the indicator function for condition A and $\theta(|\mathbf{x}|, \tau)$ is the τ^{th} biggest element of $|\mathbf{x}|$. The set $\mathcal{K}$ is a canonical simplex with projection $\mathcal{P}_{\mathcal{K}}(k)$ given by

$$P_{\mathcal{K}}(k)(i) = \max(0, k(i) - \sigma), \tag{3.8}$$

where σ is the unique solution of $1^T P_{\mathcal{K}}(k) = 1$. Both $P_{\mathcal{X}}$ and $P_{\mathcal{K}}$ can be computed in linear time using selection algorithms [31, 35].

The step sizes $\alpha_{\mathbf{x}}, \alpha_k$ are chosen by backtracking line search from an initial guess.

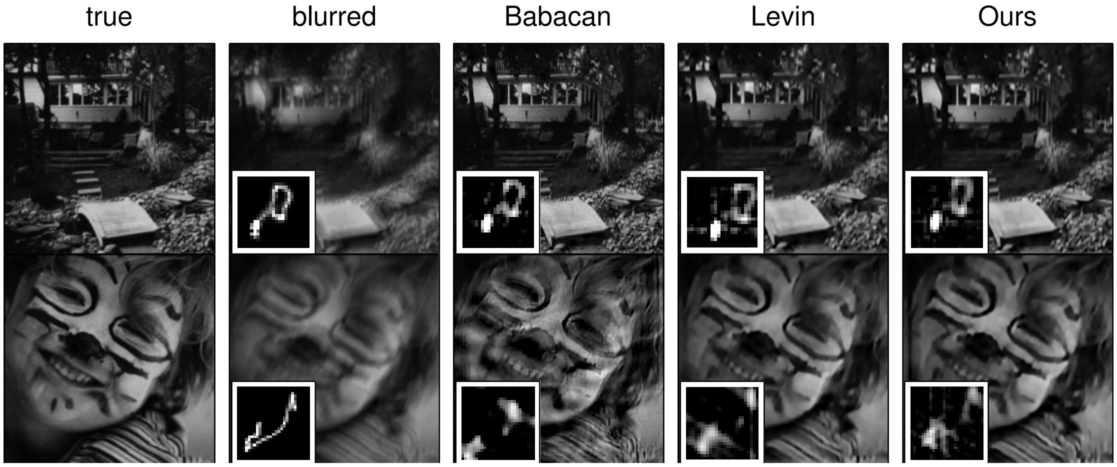

Figure 3.2: Sample results from our method, [7, 72] on the benchmark set of [71]. True and recovered kernels inset.

In the $\mathbf{x}$ subproblem our initial guess is

$$\alpha_{\mathbf{x}} = \frac{(k * g_{\mathbf{x}})^T \mathbf{r}}{(k * g_{\mathbf{x}})^T (k * g_{\mathbf{x}})}, \tag{3.9}$$

which is optimal in the sense that it solves the problem $\min_\alpha L(k, \mathbf{x} - \alpha g_{\mathbf{x}})$. This aggressive step size was chosen over several alternatives, as it was the most effective for securing good edge support estimates. In the k subproblem we use the spectral projected gradient (SPG) method [12]; in the first iteration $\alpha_k = 1$, and in subsequent iterations we use the Barzilai-Borwein step size

$$\alpha_k = \frac{(g_k - g_k^{old})^T (g_k - g_k^{old})}{(g_k - g_k^{old})^T (k - k^{old})} \tag{3.10}$$

where g_k^{old} and k^{old} denote the values of g_k and k at the previous SPG iteration.

3.3 Implementation

We implemented our method in MATLAB by modifying the code of [66], which uses a similar strategy of alternating minimization with multiscale seeding. The full-resolution kernel size was set to 35×35 for all experiments. The initial stage of the multiscale algorithm downsamples $\mathbf{y}$ by a factor of $5/35$ in each direction, so that the kernel is of size 5×5, and each upsample cycle increases the size of $k, \mathbf{x}$, and $\mathbf{y}$ by a factor of roughly $\sqrt{2}$ until full resolution is reached. The parameters of the core single-scale algorithm from §3.2 were set to $\beta_0 = 0.15$, $\gamma = 1.10$, $I_b = 20$, $I_s = 10$,

$I_{\mathbf{x}} = 1$, $I_k = 6$. We do 30 iterations of the alternating projected gradient method for all scales except the final, full-resolution scale, which uses 180 iterations. Non-blind deconvolution with the estimated kernel was performed using the method of [70], using the parameter settings chosen in the code for [72].

3.4 Numerical experiments

In [71] a test set of 32 blurry images with known ground truth was created for benchmarking blind deconvolution methods. Each blurry image was formed by taking a picture of a sharp image with a camera that shook in-plane, and bright points outside the image were used to obtain ground truth blur kernels. A total of 32 blurry images were formed by blurring 4 sharp images on 8 different shake trajectories. This test set has become the *de facto* standard for objectively comparing different methods.

We ran our algorithm on this test set and compared its performance against the methods of [72] and [7]. We compare against these methods because they have published implementations which match or exceed the performance of the state-of-the-art methods in [24, 38, 66, 93], and we know of no methods that outperform [72] and [7] on this test. We use the squared error metric $\mathrm{SSE}(x) = \sum_i (x(i) - x^{\mathrm{true}}(i))^2$ to measure performance and note that results using the ratio metric of [71] are similar. Results for [72] were taken from files included with their published implementation, while results for [7] were generated by running their online test script using the log prior, which was the best in their experiments.

Our experiments were performed in MATLAB 2011b on an Intel Quad Core Xeon 2.2 GHz Mac Pro. Our method's kernel estimation step took $45 - 60$ seconds per image, and deconvolution took 15 seconds. The other methods took $45 - 240$ seconds for kernel estimation, and their computation time depended strongly on kernel size. The difference is mostly due to our use of cheap SPG iteration rather than quadratic programming in the k step, and also because $\mathcal{P}_{\mathcal{K}}$ and $\mathcal{P}_{\mathcal{X}}$ make k and $\mathbf{x}$ sparse, enabling $k * \mathbf{x}$ to be computed faster.

Sample results on the benchmark are shown in Fig. 3.2. Our method and [72] perform very similarly on both images. On the house image [7] is very sharp and by far the best, but it suffers from severe artifacts on the boy image. Fig. 3.3 summarizes the full-benchmark performance of the three methods using cumulative error histograms. The curves for our method and [72] are largely similar. The curve for [7] is above ours and [72] for about half of the images, but it flattens out below 85% while the others plateau at 100%. This is because method [7] struggled on the boy images. We note

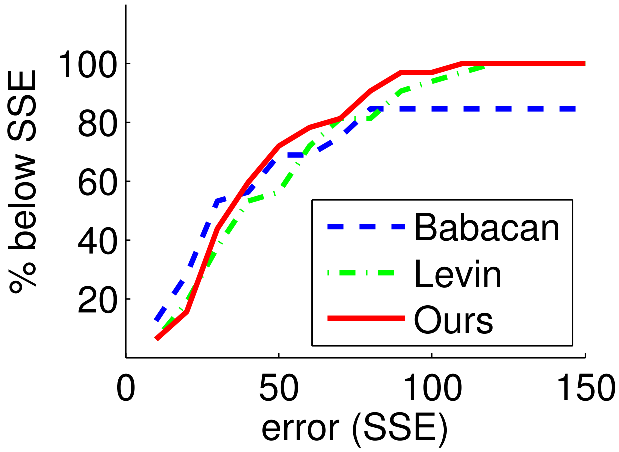

Figure 3.3: Cumulative deblurring performance our method, [72], and [7] on the 32 image test set of [71]. The vertical axis is the percentage of the 32 runs having at most a given SSE.

that the results reported in [7] for this test set are better than those obtained in our run of their code, although we ran it without any modification. The authors of [7] state that their code is a simplification of what was used to generate the reported results. While there may be a more sophisticated version of their code that outperforms ours, our method competes with the available state of the art.

3.5 Conclusion

We have proposed a blind deconvolution method in which the blur kernel is estimated by incremental sparse edge approximation. A rough global blur kernel is first estimated from only the strongest edges in the image, then it is refined as we allow the image edge map to gradually become less and less sparse. Ours is the first simple, fast joint MAP method to match the state-of-the-art kernel MAP methods in [7, 72] on an objective benchmark. The success of the methods in [66] and this paper suggest that the downsides of ideal joint MAP described in [71] can be robustly avoided without resort to a more complex kernel MAP estimation.

Our method can be improved and extended. The edge sparsity relaxation schedule we use is conservative, and a more adaptive schedule could make the method faster. Our initialization of the edge map sparsity does not take noise into account, and may need to be modified for very noisy images. Extension to nonuniform blur models, nonquadratic likelihoods, and fast parallel or GPU implementations are possible. The speed of our kernel estimation may make it useful as an input to high-quality non-blind methods such as [7].

CHAPTER 4

Stray Light Correction for STEREO/EUVI

4.1 Introduction

Extreme ultraviolet (EUV) solar images play a major role in efforts to resolve long-standing questions about the corona. EUV intensities are directly related to the coronal plasma temperature and density through the differential emission measure [17], and can even be used to compute global 3D reconstructions of the corona [39]. These data products provide powerful empirical constraints on the physics of the corona and solar wind [56, 59, 104].

A basic assumption of quantitative EUV analysis is that telescopes form an ideal geometric image: that is, each pixel's intensity is the sum of plasma emissions along that pixel's geometric line of sight through the corona. All existing EUV telescopes violate this assumption because a significant fraction of incoming light strays from the path predicted by geometric optics. The principal causes of this *stray light* are diffraction by the entrance aperture and scattering off of microrough mirror surfaces. Stray light is most significant in faint regions of the corona, where it can form a significant fraction of the observed intensity [34,94]. This paper's immediate objective is correction of stray light in the two EUVI instruments aboard the STEREO mission's Ahead (A) and Behind (B) spacecraft, which we refer to EUVI-A and B. Its broader objective is to put EUV stray light correction on the path to becoming a theoretically justified, empirically rigorous, and scientifically reliable tool applicable to SOHO/EIT, TRACE, SDO/AIA, and any future spaceborne EUV telescopes.

The distribution of stray light in a telescope is governed by its *point spread function* (PSF), which is the image of a unit intensity point source. Most of the PSF's intensity,

A version of this chapter will be submitted for journal publication with co-authors Richard A. Frazin, Anna C. Gilbert, and Alfred O. Hero III. Preliminary work on stray light correction for STEREO-B/EUVI was published with the same co-authors in [94].

or mass, is concentrated in the core, which has a subpixel width in EUVI [55]. The remainder is scattered into the *wings*, which decay slowly and extend hundreds of pixels from the core. Stray light contamination is modeled by convolution of the ideal geometric image with the PSF, and the correction process is called *deconvolution* [99]. When the PSF is unknown or partially known, it must be determined from the images themselves. This is called *blind* or *semiblind* deconvolution and is much more difficult [21].

Laboratory characterization of EUV PSFs is impossible due to the lack of a sufficiently strong EUV source, but considerable information about them can be extracted from in-flight observations. Diffraction by the entrance aperture can be characterized by analyzing the diffraction orders that appear around solar flares [43], but diffuse scatter due to mirror microroughness is not easily observed in flare images. The best information about diffuse scatter comes from transiting bodies that emit negligible EUV radiation, so whatever they appear to emit must have been scattered from the corona. Stray light levels were estimated in EIT using a transit of Mercury in [5]. The stray light's subtle anisotropy and its correlation with the pupil geometry were noted, but a PSF was not determined. DeForest *et al* used a transit of Venus to determine an isotropic Lorentzian-type PSF model for diffuse scattering in the TRACE instrument [34]. A similar method was used to obtain PSFs for SDO/AIA in [88].

In [94] the authors performed semiblind deconvolution of a lunar transit to determine an anisotropic generalized power law PSF for EUVI-B 171 Å. The PSF parameters and ideal images were treated as unknowns to be determined simultaneously from lunar transit images, leading to a multiframe semiblind deconvolution problem. This was solved by a nonlinear least squares optimization enforcing the constraint of zero intensity in the lunar disk of each ideal image. The first estimates of systematic error in stray light corrected images were obtained using a novel empirical analysis. To date, this is the only work to (1) treat the ideal, scatter-free transit image as an unknown rather than approximating it by heuristic manipulation of the observed image; (2) explicitly account for photon and CCD noise during PSF determination; and (3) provide error bars for the corrected images.

In this work, we obtain PSFs and systematic error estimates for all bands of EUVI-A and B and present some dramatic and unexpected effects of stray light correction on EUVI images. To lay a foundation for rigorous future work, we describe the ideal optical model, compare it with the very approximate model implicit in existing work (including our own), and suggest steps towards narrowing the gap between the two. Artifacts in the deconvolved images are traced to possible shortcomings of our model

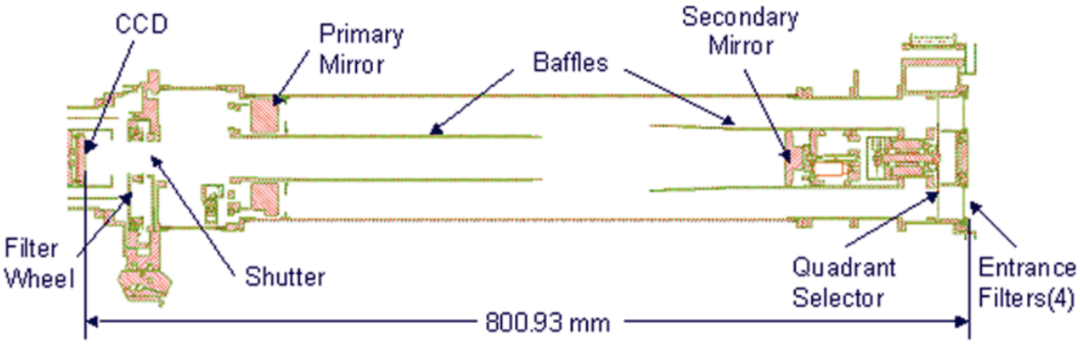

Figure 4.1: Cross section of the EUVI instrument from [55].

and suggest directions for future work.

4.2 EUVI Imaging Forward Model

4.2.1 The EUVI instrument

A cross section of each EUVI instrument is shown in Fig. 4.1. The four filter band telescopes housed by each EUVI have distinct optical paths, but each has the geometry depicted in Fig. 4.2. Light enters the primary aperture through an aluminum foil *rejection filter* which blocks almost all non-EUV light. This filter is supported by an opaque wire mesh which diffracts incoming light. The light then passes through an aperture mask slightly above the primary mirror, which is considered the instrument pupil. Next, the light is focused by primary and secondary mirrors with multilayer coatings optimized for maximum reflectivity at the selected wavelength. Near the focal plane the light passes through a backup rejection filter and is detected by a 2048×2048 back-thinned CCD array with a spatial scale of $\Delta\theta_p = 1.59$ arcsec/pixel. The wire mesh supporting the backup filter shadows the CCD slightly, modulating the image by a grid pattern with a 6% peak-to-peak amplitude. The grid pattern can be corrected by flat fielding, and the EUVI instrument team has provided flat fields for all bands except EUVI-B 284 Å, where the process is complicated by a stray light leak (J.-P. Wuelser, *private communication*).

Scattering in EUVI is primarily due to diffraction through the mesh F1 supporting the primary rejection filter, and nonspecular reflection due to microirregularities in the surfaces of the primary and secondary mirrors P1 and P2. The meshes for 171 and 195 Å are coarse and diffract little light, but those for 284 and 304 Å are fine and diffract much more. The PSF is assumed to be independent of position in the

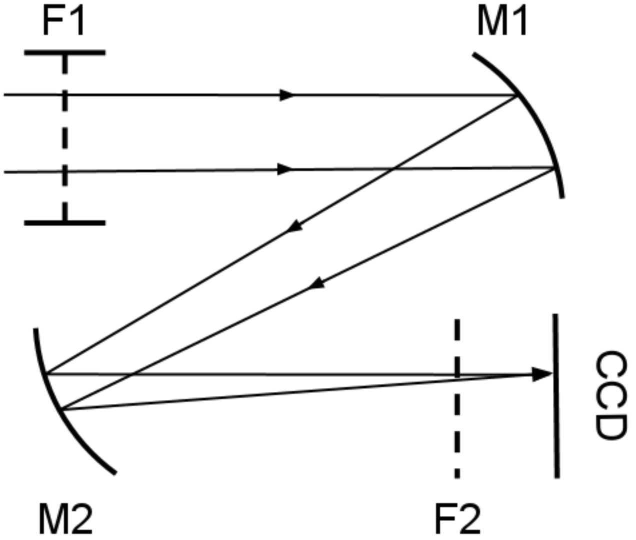

Figure 4.2: Schematic optical diagram for each EUVI filter band (not to scale). Light enters through an aluminum foil filter supported by a wire mesh (F1), is focused by the primary and secondary mirrors (M1 and M2), then passes through a second foil supported by another wire mesh (F2) before hitting the CCD.

plane as the only known source of spatial variance is geometric aberration, which is only significant within 1-2 pixels of the core (J.-P. Wuelser, *personal communication*). Since we are only interested in determining the PSF wings, these variations are ignored here.

4.2.2 PSF model

Recall that, given functions $a(x)$ and $b(x)$ defined on the plane $\mathbb{R}^2$, their convolution is defined as

$$(a * b)(x) = \int_{\mathbb{R}^2} a(x - x')b(x')dx'. \tag{4.1}$$

Our model PSF h is the convolution of two components, h^g and h^m:

$$h = h^g * h^m. \tag{4.2}$$

The *mesh component* h^g represents scattering by the grid of wires that form the mesh supporting the primary rejection filter, and is calculated by Fraunhofer diffraction theory. The *empirical component* h^m represents scattering due to mirror microroughness and other effects, and was obtained by making empirical modifications to a model from the EUV mirror scattering literature. The h^m component depends on parameters φ which are determined by semiblind deconvolution, and we write h^m_φ when we wish to make this dependence explicit. Similar multi-component convolution models have been proposed for other EUV telescopes [76, Eq. (12)]. This model differs from the model of [94] in two regards: the formula for h^m is changed, and the component h^p representing Fraunhofer diffraction from the quarter annulus pupil is omitted. The differences are discussed further in §4.2.2.2.

In Appendix 4.7.1 we show that, if the effects of the pupil and Fresnel propagation between optical elements are neglected, (4.2) can be derived from a reasonable Fourier optics model of EUVI. The accuracy lost by neglecting these effects is unclear, so the derivation serves mainly to point out a gap between theory and practice in need of further study.

4.2.2.1 Mesh component

To derive a functional form for the mesh component PSF h^g, we assume that the mesh is a grid of regularly spaced perpendicular lines of width w and spacing s. Let $x \in \mathbb{R}^2$ and $x' \in \mathbb{R}^2$ be the coordinates of the focal plane with axes parallel to the CCD principal axes and the mesh wires respectively, in units of EUVI pixels. These are related by $x = R_{\theta_g} x'$, where R_{θ_g} is a rotation matrix and θ_g is the rotation angle of the mesh relative to the CCD. Let $\xi' \in \mathbb{R}^2$ denote coordinates of the pupil plane with axes parallel to the mesh wires. We treat the wire mesh as an infinite, regular grid composed of wires with transmittance $1 - a(v)$, where $a(v)$ is the fraction of light blocked by a given wire as a function of perpendicular distance v from the wire midpoint. The mesh transmittance function is then

$$G(\xi') = \prod_{k=1,2} \left(1 - \sum_{j_k \in \mathbb{Z}} \delta(\xi'_k - s j_k) * a(\xi'_k) \right), \tag{4.3}$$

where $\mathbb{Z}$ denotes the integers. The associated Fraunhofer diffraction pattern is given by $h^g(x') \propto |\hat{G}(\tau x')|^2$, where $\tau = \Delta\theta_p/\lambda$ and λ is the peak wavelength for the filter band (Appendix 4.7.1). The resulting formula for h^g is a weighted 2D Dirac comb with spacing $1/\tau s$:

$$h^g(x') \propto \sum_{j \in \mathbb{Z}^2} m_{j_1} m_{j_2} \delta(x' - j/\tau s), \quad \text{where} \tag{4.4}$$

$$m_k = \begin{cases} 1 - (\hat{a}(0)/s)^2 & \text{if } k = 0, \\ |\hat{a}(k/s)/s|^2 & \text{if } k \neq 0. \end{cases} \tag{4.5}$$

The weight $m_{j_1} m_{j_2}$ is highest at the origin, where $j_1 = j_2 = 0$; second highest on the two coordinate axes, where one of the two $j_k = 0$; and lowest off the axes, where both $j_k \neq 0$. In EUVI we generally have $|\hat{a}(k/s)|^2 \ll 1 - |\hat{a}(0)|^2$ for $k \neq 0$, so the coordinate axis portion of the Dirac comb accounts for most of the diffraction.

In this work we follow [34, 43] and assume a step-function transition between transparency and opacity of the wires, meaning that $a(v) = \text{rect}(v/w)$, where $\text{rect}(v)$ has the value 1 on the interval $[-1/2, 1/2]$ and 0 otherwise. In this case $\hat{a}(\hat{v}) = w\,\text{sinc}(w\hat{v})$, and Dirac comb weights are

$$m_k = \begin{cases} 1 - (w/s)^2 & \text{if } k = 0, \\ (w/s)^2 \text{sinc}^2(kw/s) & \text{if } k \neq 0. \end{cases} \tag{4.6}$$

Analysis of lunar transit data suggests that this model sometimes overestimates the higher diffraction orders, particularly in 284 Å. Other forms of $\hat{a}(v)$ may improve this situation and will be considered in future work.

The EUVI instrument team has provided values of w/s, $1/\tau s$, and θ_g for both spacecraft and all filter bands in SolarSoft. For the 171 and 195 Å bands, the values are determined from the mesh manufacturer's specifications, while for 284 and 304 Å they were estimated from flare images by measuring the spacing and brightness of diffraction orders. We were unable to improve on the given values so we used them without modification in our own model.

4.2.2.2 Empirical component

In typical EUV mirror models, only a portion of incident light is scattered, and the distribution of the scattered light is given up to scaling constants by the power spectral density (PSD) of the mirror surface height function [64]. The PSD has been directly measured for some EUV mirrors, and log-log plots of the measured PSD versus spatial frequency ξ are generally piecewise linear with a rolloff at very low spatial frequencies. A reasonable model for such a graph is a sum of generalized Lorentzians: $\text{PSD}(\xi) \propto \sum_i (\overline{\rho}_i^2 + \xi^2)^{-\beta_i/2}$, where $\overline{\rho}_i$ and β_i are constants that control the rolloff transition and decay rate [76]. Assuming this model holds and the PSF is

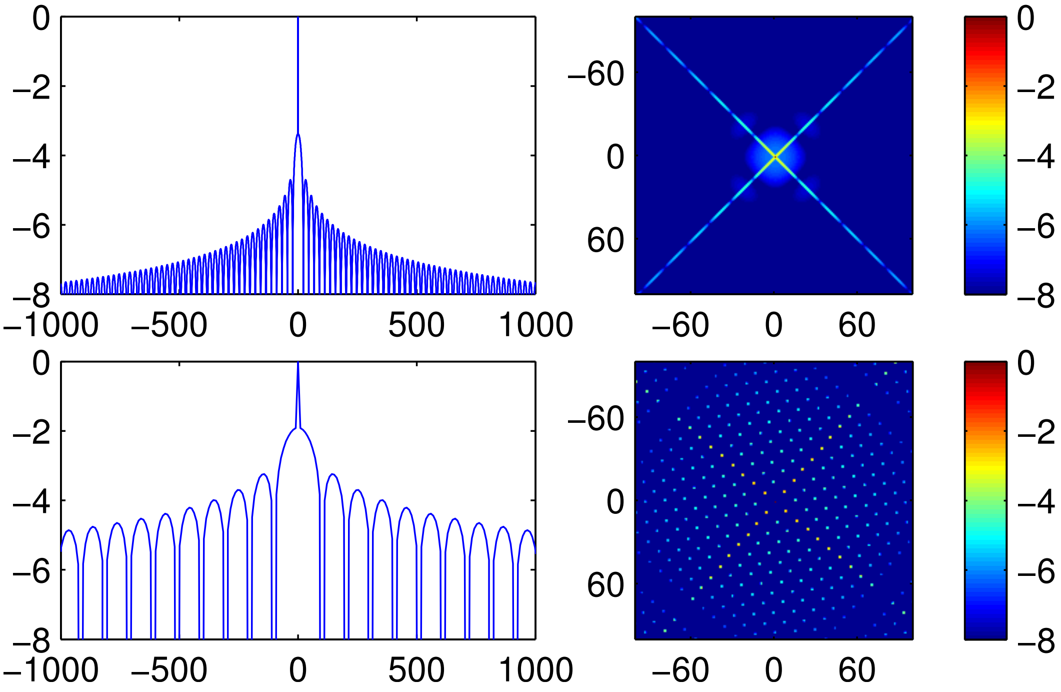

Figure 4.3: Structure of the mesh PSF h^g for 171 Å (*top row*) and 284 Å (*bottom row*). All distances are in pixels. *Left:* Log-scale plot of the k^{th} diffraction order weight, m_k, versus its displacement $k/\tau s$ from the origin. *Right:* Logarithmic colorscale plot of the core of h^g.

	λ	w/s	$1/\tau s$	θ_g (°)
A	171 Å	0.020	0.450	45.00
	195 Å	0.020	0.502	45.00
	284 Å	0.100	10.18	19.60
	304 Å	0.082	10.88	1.09
	λ	w/s	$1/\tau s$	θ_g (°)
B	171 Å	0.020	0.450	45.00
	195 Å	0.020	0.502	45.00
	284 Å	0.100	10.18	49.33
	304 Å	0.092	10.88	2.21

Table 4.1: SolarSoft values for parameters determining the form of the mesh PSF h^g. The first column, w/s, is the ratio of the mesh wire width to the wire spacing. The second column, $1/\tau s$, represents the spacing of the Dirac comb in units of pixels. The third column, θ_g, is the rotation angle of the mesh wires relative to the CCD principal axes.

isotropic, the PSF as a function of distance r from the PSF origin is

$$p_0(r) = \alpha\delta(r) + C\sum_i(\rho_i^2 + r^2)^{-\beta_i/2}, \tag{4.7}$$

where the ρ_i are Lorentzian roll-off parameters, α represents the non-scattered light, r is distance from the origin, C is a normalization constant.

The profile formula we adopt here is a special case of the formula above. We use one Lorentzian and one constant term, which can be interpreted as a Lorentzian with infinite ρ value. It is convenient to write the constant term in the form γ/N^2, where N^2 is the number of pixels in the discrete PSF, because γ then represents the total scatter due to the constant component. We then have

$$p(r) = \alpha\delta(r) + C(\rho^2 + r^2)^{-\beta/2} + \gamma/N^2. \tag{4.8}$$

This model was suggested by our initial work with piecewise power law profile from [94], which generates a class of profiles similar to (4.7). We observed that the piecewise power law tended to exhibit a single decay exponent at intermediate values of r, and rolloffs would sometimes occur at high or low r values. By replacing the piecewise power law with a generalized Lorentzian plus a constant, similar behavior is obtained with fewer free parameters.

An isotropic PSF can be generated directly from $p(r)$, but blind deconvolution of EUVI-B images with an isotropic PSF gives poor results, particularly off the limb where large sectors of negative intensity appear. A simple way to model anisotropy is to allow the PSF to have elliptical cross sections. To turn an isotropic PSF into an elliptical one, we replace the distance $r = |x|$ in the profile with $|M_{s,\theta}^{-1}x|$, where

$$M_{s,\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} s & 0 \\ 0 & 1 \end{bmatrix} \tag{4.9}$$

is the 2×2 matrix that dilates the plane by a factor of s along a line rotated θ degrees counterclockwise from the horizontal axis. The resulting PSF is

$$h^m(x) = p(|M_{s,\theta}^{-1}x|), \tag{4.10}$$

where the constant C in the definition of p is determined by the normalization constraint $\sum_x h^m(x) = 1$.

In the preliminary work of [94], we left θ as a free parameter to be determined by blind deconvolution. Here, however, θ is fixed to a particular value suggested by a confluence of theoretical and empirical evidence. Calibration roll images reveal that stray light levels are relatively high on a diagonal axis about 45° above the horizontal in 171 and 195 Å, and −45° in 284 and 304 Å (Appendix 4.7.4). These axes coincide with the scatter expected from Fraunhofer diffraction by an ideal quarter annulus pupil. In each band, the PSF h^p associated with this diffraction has two antipodal fan-shaped lobes whose axis of symmetry coincides with the axis of heightened stray light (Appendix 4.7.5). While the contribution of h^p is far too small to explain the observed anisotropy, the coincidence points to some unmodeled physical process that characteristically scatters light in the directions predicted by h^p. For this reason we set $\theta = 45°$ in 171 and 195 Å and $-45°$ in 284 and 304 Å. This leaves h^m with 5 free parameters $\varphi = (\alpha, \rho, \beta, \gamma, s)$ to be determined by semiblind deconvolution, and we write h^m_φ when we wish to make the dependence of h^m on φ explicit. The profile $p(r)$, component h^m, and full PSF h for 171 and 284 Å are shown in Fig. 4.4.

In our previous work [94] we included h^p as a component in the model PSF h. However, the discussion above suggests that we should interpret our model as containing the effect of the pupil within h^m. Additionally, we found that empirically there was no clear benefit to including h^p. For these reasons, h^p is not included as a component of h in this work.

4.2.3 Statistical image formation model

In the next sections we determine the EUVI PSFs by a model fitting process involving EUVI image data, which are subject to photon and CCD read noise. To account for the noise, we give a statistical model of EUVI image formation. We assume the image is uncompressed and has been prepared by debiasing, despiking, flat fielding, and conversion to units of photons. Assuming these corrections are accurate, instrumental effects are limited to scatter, photon noise, and CCD noise.

We let u^{true} denote the ideal geometric image, meaning that $u^{\text{true}}(x)$ is the expected photon count observed at pixel x by an ideal instrument. The expected photon count in the actual instrument is $\langle f_{\text{phot}} \rangle = h^{\text{true}} * u^{\text{true}}$, where h^{true} is the band PSF. The actual photon count, f_{phot}, is a Poisson-distributed stochastic quantity, and the photon noise is $n_{\text{phot}} = f_{\text{phot}} - \langle f_{\text{phot}} \rangle = f_{\text{phot}} - h^{\text{true}} * u^{\text{true}}$. Histograms of dark images indicate that the CCD noise, n_{ccd}, is roughly Gaussian with standard deviation $\sigma_{\text{ccd}} \approx 1$ digital number (DN). To convert the CCD noise to units of photons, we divide it by the photoelectric gain constant γ (photons/DN). The observed image

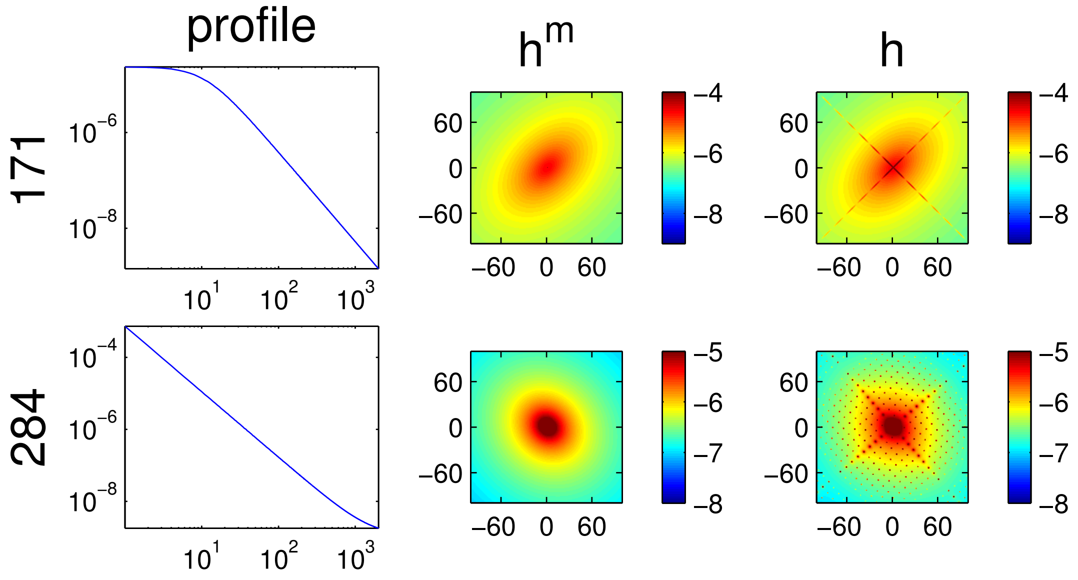

Figure 4.4: The empirical PSF h^m and total PSF h for EUVI-B, 171 and 284 Å. All distances are in pixels. *Left:* log-log plot of power-law profile function $p(r)$. *Center:* Logarithmic colormap of h^m core. *Right:* core of final PSF $h = h^g * h^m$.

f is then given by

$$f = h^{\text{true}} * u^{\text{true}} + n, \tag{4.11}$$

where $n = n_{\text{phot}} + n_{\text{ccd}}/\gamma$.

The debiasing and despiking corrections in SolarSoft did not meet the unusually stringent requirements of stray light correction, and had to be replaced or augmented by custom preparation steps described in Appendix 4.7.2. After preparation the images conform to (4.11) well enough for PSF determination, although error analysis requires consideration of subtler effects to be described later.

Our model ignores Fano noise, dark current, and charge spreading, as these effects do not impact the stray light distribution at a scale of dozens of pixels. It also ignores the effects of compression, which is a significant limitation since most EUVI images are compressed on-board by the wavelet-based ICER algorithm [63]. We use only uncompressed images for PSF determination, as needed information about the large-scale properties of low intensity regions is sometimes lost during quantization of low-pass wavelet coefficients. The PSFs we obtain can be used to correct stray light in ICER compressed images, but accuracy in faint regions can be degraded.

4.3 Method

4.3.1 Discrete convolution and deconvolution

Before explaining how the PSFs were determined, we require appropriate computational procedures for convolution and deconvolution of EUVI images and PSFs. We begin by setting a convenient array indexing notation which places the origin in the image center. The elements of an $N \times N$ array are indexed by the vector $x = (x_1, x_2) \in \mathbb{Z}_N^2$, where

$$\mathbb{Z}_N = \{-\lfloor N/2 \rfloor, -\lfloor N/2 \rfloor + 1, \ldots, -\lfloor N/2 \rfloor + N - 1\} \tag{4.12}$$

is the set indexing the rows and columns, and $\lfloor N/2 \rfloor$ is the largest integer less than or equal to $N/2$.

Given $N \times N$ arrays h and u, their discrete convolution is obtained by replacing the convolution integral with a summation:

$$(h * u)(x) = \sum_{x' \in \mathbb{Z}_N^2} h(x - x')u(x'). \tag{4.13}$$

If h is a PSF and u is an image, $h * u$ represents the blurring of u by h. This sum involves indices $x - x' \notin \mathbb{Z}_N^2$, so we assume $h(x - x') = 0$ in this case, meaning that the convolution is *zero-padded.* This assumption is reasonable for EUV PSFs and images because both take very small values near the array boundary.

Discrete convolutions can be represented by using the discrete Fourier transform (DFT), which is efficiently computable via the fast Fourier transform (FFT) algorithm. The DFT of an array u will be denoted $\hat{u}$ or $\mathcal{F}[u]$, and within the scope of this paper will be defined as

$$\hat{u}(\xi) = \sum_{x \in \mathbb{Z}_N^2} u(x) \exp[-2\pi i \langle x, \xi \rangle / N] \quad \text{for } \xi \in \mathbb{Z}_N^2, \tag{4.14}$$

where $\langle x, \xi \rangle$ is the dot product. (This formula is related to the standard DFT formula by a shift of the pixel indices.) The convolution of h and u with *periodic* boundary conditions, allowing blur to wrap from one side of the image to the other, can easily be computed using the formula $\mathcal{F}^{-1}[\hat{h} \cdot \hat{u}]$, where $(\hat{h} \cdot \hat{u})(\xi) = \hat{h}(\xi)\hat{u}(\xi)$ represents pointwise multiplication. This periodic convolution formula is very commonly used in the image processing literature, but is unsuitable for EUV images, as the unphysical wrapping can cause large-scale artifacts off the limb.

Fortunately, zero-padded convolutions can be computed using a padded FFT. Given a $N \times N$ array a, let a_{pad} represent the $2N \times 2N$ array obtained by padding of a with zeros: for $x \in \mathbb{Z}_{2N}^2$,

$$a_{\text{pad}}(x) = \begin{cases} a(x) & \text{if } x \in \mathbb{Z}_N^2 \\ 0 & \text{otherwise.} \end{cases} \tag{4.15}$$

Then $h * u$ is obtained by padding h and u, taking their periodic convolution, and clipping off the padding:

$$h * u = \text{Clip}\left\{\mathcal{F}^{-1}\left[\hat{h}_{\text{pad}} \cdot \hat{u}_{\text{pad}}\right]\right\}, \tag{4.16}$$

where the Clip operator returns the $N \times N$ subarray indexed by $\mathbb{Z}_N^2$.

Implementing zero-padded deconvolution with FFTs is slightly more complicated than convolution because a finitely supported inverse kernel h^{-1} does not exist in general. The most accurate deconvolution is obtained by solving the linear system $h * u = f$, but there is a convenient and accurate approximation for EUV imaging purposes. We define an approximate inverse kernel $h^{-1} = \mathcal{F}^{-1}[1/\hat{h}_{\text{pad}}]$ and its approximate convolution with f as

$$h^{-1} * f \approx \text{Clip}\left\{\mathcal{F}^{-1}\left[(\hat{f}_{\text{pad}}/\hat{h}_{\text{pad}})\right]\right\}. \tag{4.17}$$

This approximation is most accurate when f and h decay to zero near the image boundary, which is generally the case for EUVI images and PSFs. When applied to EUVI images, this formula's output differed negligibly from the solution of $h * u = f$ obtained by conjugate gradients [101], so (4.17) was adopted as the standard numerical deconvolution method in this work.

In most deconvolution applications there are spatial frequencies ξ such that $\hat{h}_{\text{pad}}(\xi)$ is either zero or very small, so the division by $\hat{h}_{\text{pad}}$ in (4.17) would be either undefined or greatly amplify the image noise. In our case, however, this problem never occurs because the PSF core is represented by a single pixel at the origin. This means that most of the total mass of h is found at the origin, which prevents $\hat{h}_{\text{pad}}$ from becoming too small. In particular, it can be shown mathematically that $1 \geq |\hat{h}_{\text{pad}}(\xi)| \geq 2h(0) - 1$. Eq. (4.6) and Table 4.1 imply that $h^g(0) > 0.8$ for all bands of EUVI-A and B, while Table 4.2 says that $h^m(0) = \alpha > 0.8$ for all bands of EUVI-A and B. This implies that the discrete convolution $h = h^g * h^m$ has $h(0) > 0.8^2 = 0.64$, so $|\hat{h}_{\text{pad}}(\xi)| > 0.28$ for all ξ. If we modeled the PSF core realistically, rather than as a single pixel, this would no longer be the case and it would be necessary to regularize the deconvolution [99].

4.3.2 Preparations for PSF determination

We determined the EUVI PSFs in two stages. First, the EUVI-B PSFs were determined using lunar transit data. Second, EUVI-A PSFs were determined using EUVI-A and EUVI-B images from the early STEREO mission. In both cases the PSFs were determined by solving semiblind deconvolution problems derived from (4.11), where the PSF h^{true} is assumed to come from the model $h = h^g * h^m_\varphi$, and the image u^{true} obeys certain side constraints.

Performing maximum likelihood inference with (4.11) is computationally challenging because of the Poisson-Gaussian noise $n = n_{\text{phot}} + n_{\text{ccd}}/\gamma$. In [94] an Anscombe transform was applied to (4.11) to make the noise more Gaussian, but the nonlinearity of the transform can cause computational troubles and possibly statistical bias. In this work we take a simpler approach: we first deconvolve h^g from the image f using (4.17), then apply a 4×4 binning to the image, reducing the number of pixels by a factor $N_b = 16$, and call the resulting image f^m. After this operation is completed, we also apply 4×4 binning to f, so that

$$f = h^{\text{true}} * u^{\text{true}} + n \tag{4.18}$$

$$f^m \approx h^{m,\text{true}}_\varphi * u^{\text{true}} + n, \tag{4.19}$$

where all arrays are now 512×512. (In the second equation we have implicitly assumed that $(h^g)^{-1} * n$ is roughly n because the noise amplification from deconvolution with h^g is modest.) The mesh PSF h^g is deconvolved from f before binning because, as a modulated Dirac comb, it has pixel-scale structure. In contrast, h^m_φ varies only on scales of dozens of pixels, so it is not much affected by binning.

To see why 4×4 binning makes n roughly Gaussian, note that the CCD noise variance in the binned image is $\langle (n_{\text{ccd}}/\gamma)^2 \rangle = N_b \sigma^2_{\text{ccd}}/\gamma^2 \approx 16$, since $N_b = 16$ and $\gamma, \sigma_{\text{ccd}} \approx 1$. When $\langle f \rangle \ll 16$, the photon noise variance $\langle n^2_{\text{phot}} \rangle = \langle f \rangle \ll 16$, so the Gaussian noise n_{ccd} is the dominant contribution to n. When $\langle f \rangle \gtrsim 16$ we enter the high count regime where Poisson noise becomes roughly Gaussian, which means that the sum n is the sum of two independent Gaussians. In either case, n reasonably modeled as Gaussian, and has a variance of $\langle n^2 \rangle = \langle f \rangle + N_b \sigma^2_{\text{ccd}}/\gamma^2$.

A numerical estimate σ^2 of the variance $\langle n^2 \rangle$ is required to define the maximum likelihood problem that determines the PSFs. Since $\langle f \rangle$ is unobservable, use the approximation $\langle f \rangle \approx f$ and set $\sigma^2 = f + N_b \sigma^2_{\text{ccd}}/\gamma^2$. Since the signal-to-noise ratio is

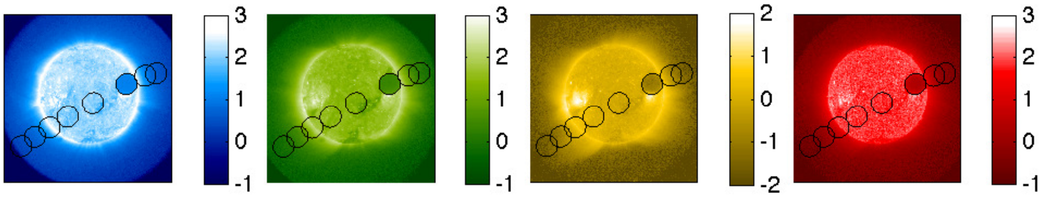

Figure 4.5: A lunar transit image from the series of 8 uncompressed images in each EUVI band (units of $\log_{10}$ ph/sec). The lunar disk moves from lower left to upper right over the course of 16 hours. Black circles show the positions of the lunar disk during the transit. In these images, the 6^{th} of each series, an active region can be seen just south of the lunar disk. The path is slightly bent due to shifts in pointing during the transit.

$\langle n^2\rangle^{1/2}/\langle f\rangle \approx \langle f\rangle^{-1/2}$, which approaches 0 in the high count limit, this approximation is good except in the faintest regions of the image. Empirically we have found that small alterations in σ^2, for example doubling or tripling σ_{ccd}, do not have much impact on the PSF determined. Thus it seems unlikely that the approximation $f \approx \langle f\rangle$ has much impact on the result.

To avoid computational difficulties associated with unreasonable PSFs, we require the variables in φ to stay within the bounds defined by the following set:

$$\begin{aligned}\Phi = \{(\alpha,\rho,\beta,\gamma,s) : \tfrac{1}{2} \le \alpha \le 1,\ \rho \ge 0,\ 0 \le \beta \le 5, \\ 10^{-8} \le \gamma \le 10^{-1},\ \tfrac{1}{5} \le s \le 5\}.\end{aligned} \tag{4.20}$$

4.3.3 Determining EUVI-B PSFs from the lunar transit

We determined φ for each band using a series of 8 images $f_1,\ldots,f_8$ from the Feb. 25, 2007 lunar transit, which is depicted in Fig. 4.5. We constrain the ideal images u_i to be zero on the lunar disk pixels Z_i: $u_i(Z_i) = 0$ for $i = 1,\ldots,8$. The lunar disk was identified by detecting its edge pixels with gradient thresholding, then fitting a circle to the detected edge pixels. The PSFs are obtained by seeking an approximate solution to the following nonlinear least squares problem, which itself approximates the maximum likelihood problem under (4.19):

$$\begin{aligned}&\underset{\varphi\in\Phi,\ \{u_i\}}{\text{minimize}} \quad \sum_{i=1}^{8}\sum_{x\in S_f}\left(\frac{(h_\varphi^m * u_i)(x) - f_i^m(x)}{\sigma_{f_i}(x)}\right)^2 \\ &\text{subject to} \quad u_i(Z_i) = 0 \text{ for all } i,\end{aligned} \tag{4.21}$$

where $\sigma_{f_i}^2 = f_i + N_b \sigma_{\text{ccd}}^2 / \gamma^2$ and S_f is the set of unvignetted pixels. Even with the f_i^m and u_i binned to 512×512, this problem has over 2 million variables. Problems of this size can be solved directly by quasi-Newton and truncated Newton-type methods [82] but convergence can be slow due to the problem's size and ill-conditioned Hessians [95].

To reduce the problem's size, we use an approximate variant of the variable elimination technique described in [45]. In the variable elimination technique, a formula is obtained for the optimal u_i in (4.21) given a fixed value of φ. This formula for the minimizer is then substituted in for each u_i in (4.21). Each u_i appears in only the i^{th} term of (4.21), and thus can be found by solving

$$\begin{array}{ll} \underset{u_i}{\text{minimize}} & \displaystyle\sum_{x \in S_f} \left(\frac{(h_\varphi^m * u_i)(x) - f_i^m(x)}{\sigma_{f_i}(x)} \right)^2 \\ \text{subject to} & u_i(Z_i) = 0. \end{array} \tag{4.22}$$

To approximate the optimal u_i we deconvolve h_φ^m from f_i^m and set the lunar disk to zero:

$$u_{i,\varphi}(x) = \begin{cases} ((h_\varphi^m)^{-1} * f_i^m)(x) & \text{if } x \in S_f \text{ and } x \notin Z_i, \\ 0 & \text{otherwise} \end{cases} \tag{4.23}$$

Plugging this formula into (4.21) results in an optimization over φ alone:

$$\underset{\varphi \in \Phi}{\text{minimize}} \quad \sum_{i=1}^{8} \sum_{x \in S_f} \left(\frac{(h_\varphi^m * u_{i,\varphi})(x) - f_i^m(x)}{\sigma_{f_i}(x)} \right)^2 . \tag{4.24}$$

In contrast to traditional variable elimination, (4.24) is not precisely equivalent to (4.21) because $u_{i,\varphi}$ is only an approximation of the optimal u_i in (4.22). For our purposes the approximation seems to work quite well. To avoid boundary effects and speed up computation we found it convenient to sum over only the lunar disk pixels Z_i, where most of the residual $h_\varphi^m * u_{i,\varphi} - f_i^m$ is concentrated. The final problem we sought to solve was then

$$\underset{\varphi \in \Phi}{\text{minimize}} \quad \sum_{i=1}^{8} \sum_{x \in Z_i} \left(\frac{(h_\varphi^m * u_{i,\varphi})(x) - f_i^m(x)}{\sigma_{f_i}(x)} \right)^2 . \tag{4.25}$$

Eq. (4.25) was solved using `lsqnonlin`, a MATLAB routine for nonlinear least squares. This method requires only a function that takes an input φ and outputs a vector composed of the residual values $r_{i,\varphi}(x) = ((h_\varphi^m * u_{i,\varphi})(x) - f_i^m(x))/\sigma_{f_i}(x)$

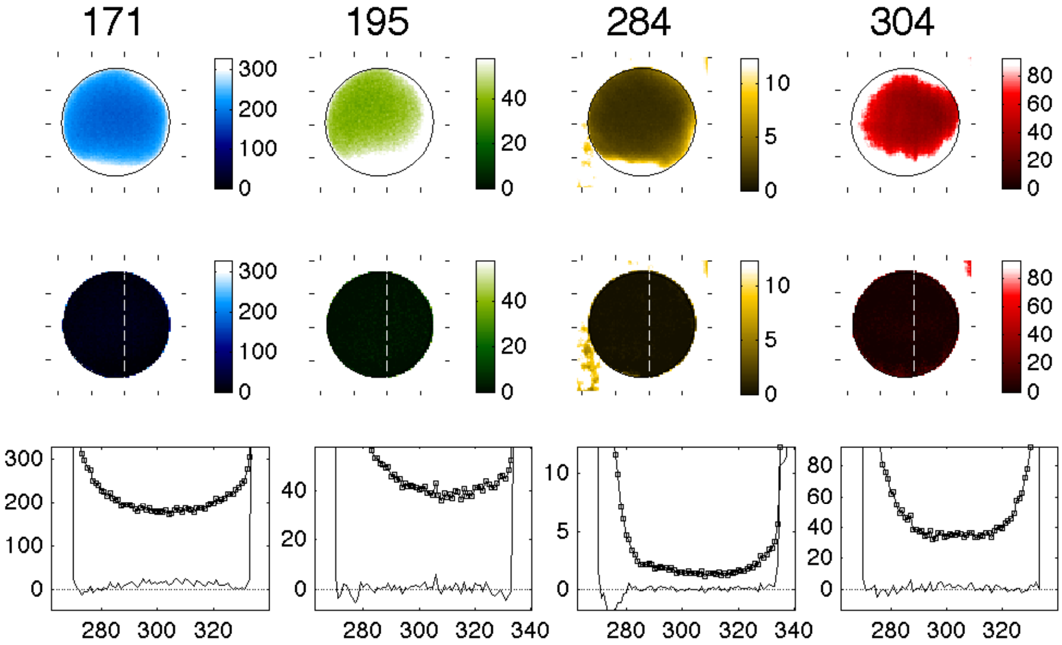

Figure 4.6: Lunar transit images before stray light correction (*top row*) and after (*middle row*), in units of ph/sec. All images are 4 × 4 binned, and tick marks are given every 20 binned pixels. The color scale has a low upper limit to make the stray light visible. The lunar disk is outlined, and a vertical dotted line is drawn through the corrected images. *Bottom:* Intensities on the dotted line before stray light correction (*squares*) and after (*solid line*).

for all images i and all $x \in Z_i$. Derivatives of the residuals were approximated by finite differences. A high-precision solution was obtained from a laptop within a few minutes for each band.

The parameters φ determined for each 512×512 PSF were slightly modified to obtain a full 2048×2048 PSF. Specifically, ρ was multiplied by the binning factor 4, and α was slightly reduced to ensure that the origin pixel of the 512×512 PSF has the same mass as the corresponding 4×4 square in the 2048×2048 PSF. The other parameters are scale-invariant and need no modification. In Table 4.2 we report the resulting parameter values φ for each of the four bands.

To see the effect of stray light correction on lunar transit images, we generated a PSF for each band using the model $h = h^g * h^m_\varphi$ and the values of Table 4.2, and deconvolved it from the lunar transit images f_i. In Fig. 4.6, we examine the lunar disk from the 6^{th} image before deconvolution (*top row*) and after (*middle row*). An active region immediately south of the lunar disk scatters a large amount of light into it, making this image especially challenging for deconvolution. Despite this the disk is dark in every case. In the bottom row, intensities are plotted along a vertical line through the original and deconvolved images. The plots show a consistent and dramatic reduction in lunar disk intensity after deconvolution.

The most noticable deconvolution artifacts are a slight positive bias in the middle of the 171 Å disk and an overcorrection, resulting in negative intensity, near the bottom of the disk in 284 Å. The reason for the bias in 171 Å is not clear, but the overcorrection in 284 Å derives from excessively heavy weights $m_{j_1}m_{j_2}$ on the point masses lying on the principal axes of the Dirac comb for h^g (see (4.4)).

4.3.4 Determining EUVI-A PSFs from early mission data

Simultaneous imaging by EUVI-A and B began on December 14, 2006, when the A and B spacecraft had a separation angle of 0.002° relative to the Sun. Separation increased rapidly in the following months, but remained below 0.05° for the rest of December. A simple calculation shows that at a distance of 1 AU, the change in line of sight from a 0.05° orbital displacement shifts the disk center by 0.5 EUVI pixels, and this is the largest possible shift on the disk. Up to a subpixel difference, then, the ideal geometric solar image was the same for EUVI-A and B during December.

To express this fact in terms of the model (4.11), we let f_A be the EUVI-A image and f_B the EUVI-B image. The image orientations are subject to pointing differences, so we coalign f_B with f_A, meaning we shift and rotate f_B so its disk center and solar North angle match f_A. The PSF for the rotated f_B is obtained by rotating the B PSF, h_B^{true}, by the same angle applied to f_B. We then have

$$f_A = h_A^{\text{true}} * u^{\text{true}} + n_A \tag{4.26}$$

$$f_B = h_B^{\text{true}} * u^{\text{true}} + n_B. \tag{4.27}$$

We obtain an estimate u for u^{true} by deconvolving the estimated EUVI-B PSF h_B from f_B, then coaligning u with f_A using bilinear interpolation. We then deconvolve h_A^g from f_A to obtain f_A^m, and bin both u and f_A^m down to 512×512. Substituting u into (4.19) we obtain

$$f_A^m \approx h_{A,\varphi}^m * u + n_A, \tag{4.28}$$

where φ now denotes the parameters for the EUVI-A PSF. These are obtained by

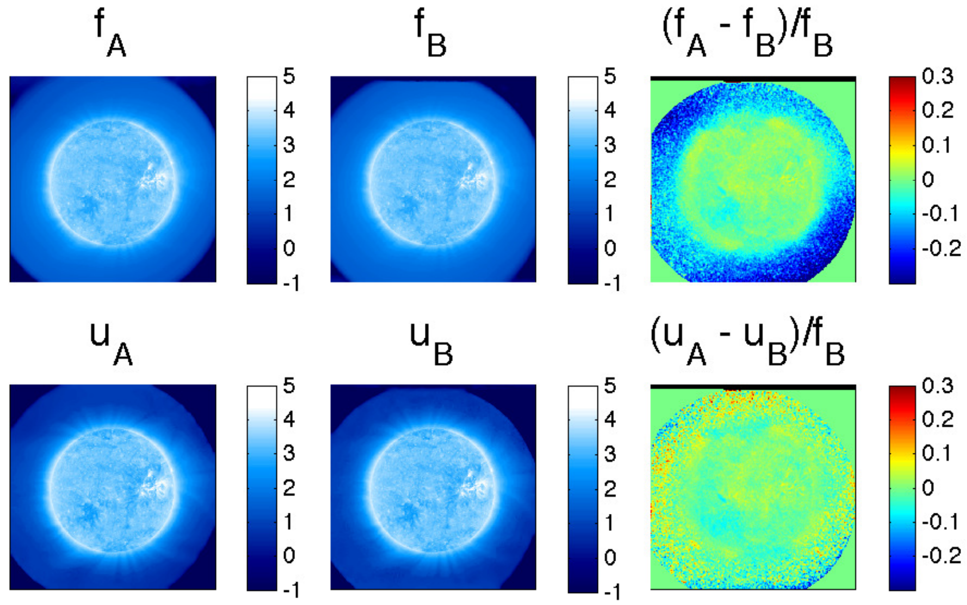

Figure 4.7: A simultaneous exposure by EUVI-A and B in the 171 Å band on Dec. 14, 2007 UTC 18:10:60 (units of $\log_{10}$ ph/sec). *Top row, left to right:* The original images f_A and f_B, and their relative difference $(f_A - f_B)/f_B$. Note that the off-limb in A is up to 30% dimmer than in B. *Bottom row, left to right:* Deconvolved images u_A and u_B, and the relative difference $(u_A - u_B)/f_B$. The off-limb discrepancy is greatly reduced.

solving

$$\underset{\varphi\in\Phi}{\text{minimize}} \quad \sum_{x\in S_f} \left(\frac{h^m_{A,\varphi} * u - f^m_A}{\sigma_{f_A}} \right)^2, \tag{4.29}$$

where $\sigma^2_{f_A} = f_A + N_b\sigma^2_{\text{ccd}}/\gamma^2$ is the noise variance estimate. The MATLAB utility `lsqnonlin` was used to solve this problem exactly as in the previous section, except the residual vector is now composed of the values of $(h^m_{A,\varphi} * u - f^m_A)/\sigma_{f_A}$. The images used were simultaneous A/B exposures in each band taken on December 14, 2006 from 17:45:00 to 17:46:30 UTC. The φ values obtained for each EUVI-A band are reported in Table 4.2.

In the left and middle columns of Fig. 4.7 coaligned 171 Å images are shown before stray light correction (*top*) and after (*bottom*). In the right column are the relative differences $(f_A - f_B)/f_B$ and $(u_A - u_B)/f_B$. Generally f_A and f_B are within ±10% of each other on the solar disk, but off the limb f_B is up to 30% brighter than f_A. In the deconvolved images u_A, u_B, the off-limb discrepancy is almost completely eliminated. Similar changes are observed in the other bands.

	λ	α	ρ	β	$\log_{10}\gamma$	s
A	171 Å	0.856	20.607	2.058	-2.334	1.622
	195 Å	0.901	19.989	2.072	-2.028	1.838
	284 Å	0.935	114.31	2.991	-2.034	1.641
	304 Å	0.952	7.266	2.111	-2.129	2.093
	λ	α	ρ	β	$\log_{10}\gamma$	s
B	171 Å	0.832	11.108	1.869	-8.000	1.546
	195 Å	0.880	5.613	1.834	-2.471	1.300
	284 Å	0.918	0.020	1.821	-2.339	1.163
	304 Å	0.907	0.000	2.117	-1.994	1.422

Table 4.2: Parameters determining 2048 × 2048 PSFs of EUVI-A and B in all four filter bands.

4.4 Error Analysis

Before our error analysis can proceed we must add to the image formation model an effect that was neglected during PSF determination. In Appendix 4.7.2, an analysis of pixel values in vignetted image regions reveals that the image bias (the pixel value in areas of zero intensity) varies on scales of dozens to hundreds of pixels. The bias variation ranges from 0.03 to 0.2 DN for the images considered in this paper, and its cause is uncertain. The bias variation is negligible for most purposes, but in very faint off-limb regions and on large spatial scales it can become a major source of error. We treat the bias variation as an additive spatially varying function, $\Delta B(x)$, and add it to (4.11) to obtain

$$f = h^{\text{true}} * u^{\text{true}} + n + \Delta B. \tag{4.30}$$

Given this formula, the total error $u - u^{\text{true}}$ in a deconvolved image can be decomposed into three components. We set $u = h^{-1} * f$ and plug into (4.30) to obtain

$$\begin{aligned} u - u^{\text{true}} &= h^{-1} * f - u^{\text{true}} \\ &= ([h^{-1} * h^{\text{true}} - \delta] * u^{\text{true}}) + (h^{-1} * n) + (h^{-1} * \Delta B) \\ &= \epsilon_{\text{psf}} + \epsilon_n + \epsilon_{\text{bv}}. \end{aligned} \tag{4.31}$$

The ϵ_{psf} term represents systematic error due to PSF inaccuracy, ϵ_n represents random error due to propagation of noise in the observed image, ϵ_{bv} represents error due to bias variation.

Our estimates of these errors, σ_{psf}, σ_n, and σ_{bv}, are computed below. These errors come from very different sources and are expected to be uncorrelated with each other, so they may be added in quadrature to estimate the total error σ_u in the deconvolved image u:

$$\sigma_u^2 = \sigma_{\text{bv}}^2 + \sigma_{\text{psf}}^2 + \sigma_n^2. \tag{4.32}$$

The random error ϵ_n has minimal correlations above the pixel scale, but the systematic errors ϵ_{psf} and ϵ_{bv} are spatially correlated on scales of dozens of pixels or more. Therefore, if filtering or binning are applied to the images, the noise and systematic errors must be treated differently. For example, ϵ_n will be reduced by a factor of about $N_{\text{av}}^{1/2}$ if a moving average over N_{av} pixels is applied, but σ_{bv} and σ_{psf} will not necessarily be reduced.

To calculate σ_{bv} we note that since $\Delta B(x)$ varies slowly with x, it is minimally affected by deconvolution, so it is reasonable to use the approximation $\epsilon_{\text{bv}} = h^{-1} * \Delta B \approx \Delta B$. We then set $\sigma_{\text{bv}} = \sigma_{\text{vc}}$, where σ_{vc}, defined in (4.48), estimates the standard deviation of the four vignetted corners' mean intensities from their grand mean. The bias variation term is not present in the error model of [94] because we had not yet discovered it.

The noise estimate σ_n is determined by calculating $\langle (h^{-1} * n)^2 \rangle$, the diagonal of the covariance matrix for $h^{-1} * n$. Given the estimate of $\langle n^2 \rangle \approx \sigma_f^2 = f + \sigma_{\text{ccd}}^2/\gamma^2$ described in §4.3, covariance propagation can be used to show that

$$\langle (h^{-1} * n)^2 \rangle = (h^{-1})^2 * \langle n^2 \rangle \approx (h^{-1})^2 * \sigma_f^2 = \sigma_n^2, \tag{4.33}$$

where $(h^{-1})^2(x) = h^{-1}(x)^2$. Other parts of the covariance matrix for ϵ_n can also be computed, but are not considered here. Deconvolution generally increases the noise level modestly and introduces mild, pixel-scale spatial noise correlations.

The estimate σ_{psf} of error due to PSF inaccuracy is obtained by an empirical analysis of lunar transit and early mission images, as detailed in the following sections. The goal of the analysis is to estimate the distribution of ϵ_{psf}/f, the PSF error as a fraction of the observed image intensity. Given this distribution we then take $\sigma_{\text{psf}} = \rho f$ as our systematic error estimate, where the scalar constant ρ is some statistic measuring the distribution's spread. This is a similar but more conservative bound than was used in [94], where it was assumed that error due to PSF inaccuracy was proportional to the magnitude of the correction: $\sigma_{\text{psf}} \propto |f - u|$. Such an estimate implies that the error is very small when the stray light correction is small, and we suspect from work with EUVI-A that this may be too optimistic.

The naïve approach of estimating the distribution of ϵ_{psf} itself is not effective because (4.31) implies that ϵ_{psf} is a filtered solar image, and as such its dynamic range in a given region depends strongly on the local intensity profile. For example, ϵ_{psf} generally has a much larger magnitude on the disk than off the limb, while the relative error ϵ_{psf}/f has a much more consistent range.

4.4.1 Reducing noise before estimation of ϵ_{psf}

The empirical analysis of ϵ_{psf} below requires that the contribution of noise be minimized in both the original and deconvolved images. Noise can be reduced at the cost of resolution by binning and spatial averaging. Reduced resolution is acceptable to us since we are interested only in the stray light distribution on the scale of dozens of pixels.

For each EUVI image used in the estimation of ϵ_{psf}, a full-size deconvolved image was computed using (4.17). Both observed and deconvolved images were then binned from 2048 × 2048 to 512 × 512, which is sufficient to make noise negligible in all but the faintest regions. These remaining faint regions generally have minimal small-scale structure, and can be denoised by a spatially adaptive averaging technique known as wavelet thresholding. This technique smoothes away small local fluctuations due to noise while preserving large local fluctuations and global trends. We apply a wavelet thresholding procedure described in Appendix 4.7.3 to all observed and deconvolved images, and in the following analyses we assume that noise is negligible.

4.4.2 Estimating ϵ_{psf} in EUVI-B

We estimate ϵ_{psf} for EUVI-B by examining lunar disk pixel values in deconvolved transit images. Since the lunar disk does not emit EUV radiation, these values represent observable errors in the deconvolved images. Underlying our analysis is the assumption that the observable errors are representative of the unobservable errors occurring in general usage of the deconvolution. While the accuracy of this assumption cannot be checked directly, the diversity of positions taken by the lunar disk during the transit (within the disk, off the limb, cutting through the limb) help to ensure that it holds.

Before proceeding, we note that all 8 uncompressed lunar transit images in each band were used to determine each band's PSF, so there is no separate data on which to perform the error analysis. An error analysis performed on the same data used to fit a model tends to underestimate the error incurred in the model's general usage, a phenomenon known as overfitting [54]. The risk of overfitting is greatest for complex models with many parameters. While our PSF model has only a few parameters, some risk may still remain.

To mitigate the risk, we instead consider images $u_i^\star$ derived from the following cross-validation (CV) procedure. For each of the 8 images f_i, we determined a PSF h_i by solving (4.25) with f_i removed from the dataset. We then calculated the stray light corrected CV image $u_i^\star = h_i^{-1} * f_i$. Since f_i was not used to fit h_i, the values of $u_i^\star$ on the lunar disk represent an independent check on the correction's effectiveness. For each of the 8 lunar transit images f_i we know that $u_i^{\text{true}} = 0$ on the lunar disk Z_i, so deviations of the denoised $u_i^\star$ from zero are due to PSF and bias variation error:

$$u_i^\star(x) = \epsilon_{\text{psf},i}(x) + \epsilon_{\text{bv},i}(x) \quad \text{for } x \in Z_i. \tag{4.34}$$

We have no way to remove $\epsilon_{\text{bv},i}$ so that $\epsilon_{\text{psf},i}$ can be estimated in isolation, so we must be careful not to consider areas where the two quantities may be comparable. The value of $\epsilon_{\text{bv},i}$ is on the order of $N_b\sigma_{\text{bv},i}$ per binned pixel, while $\epsilon_{\text{psf},i}$ is assumed to be proportional to f_i. Thus, in regions where $f_i \approx N_b\sigma_{\text{bv},i}$, we cannot safely assume that $\epsilon_{\text{psf},i}$ dominates. We found that excluding pixels where $f_i(x) < 2N_b\sigma_{\text{bv},i}$ was sufficient to prevent pathologies in the analysis, and the exclusion only affected a small fraction of pixels.

For the remaining pixels x it was assumed that PSF inaccuracy was responsible for the observed error: $u_i^\star(x) \approx \epsilon_{\text{psf},i}(x)$. For each image i, the ratio $u_i^\star(x)/f_i(x)$ was computed for all lunar disk pixels x such that $f_i(x) \geq 2N_b\sigma_{\text{bv}}$, and the resulting collection of ratios was collected into a single vector, R_B. Histograms of R_B were used to estimate the distribution of ϵ_{psf}/f for a generic image f.

In Fig. 4.8, *top row*, we show the relative error map $u_i^\star/f_i$ on a lunar transit image from each filter band. The disks shown have among the highest ratios due to an active region just south of the lunar disk. Despite this, the ratio's value (as a percentage) is generally less than 10% in each band, and less than 20% in 284 Å. The largest errors occur in a ring of four negative (blue) regions around the lower edge of the lunar disk in 284 Å. These regions are caused by overcorrection of grid diffraction associated with an active region just below the lunar disk. The PSF h^g for 284 Å implies more scatter from this active region than is actually observed, so the deconvolved image has pockets of negative values. Similar, less prominent pockets can be seen in the other bands. The overcorrection associated with h^g derives from excessively heavy weights $m_{j_1}m_{j_2}$ on the point masses lying on the principal axes of the Dirac comb for h^g (see (4.4)). Adjustment of these weights to improve the deconvolution's accuracy will be considered in future work.

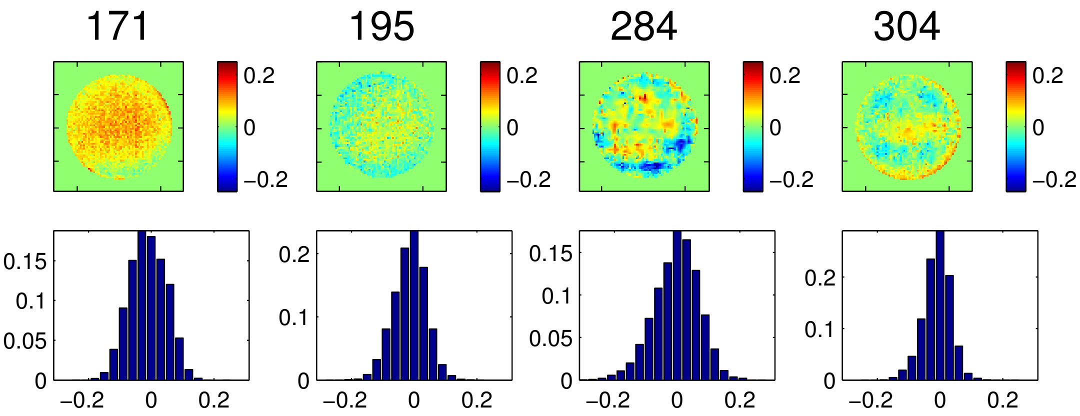

Figure 4.8: Empirical estimation of relative PSF error in EUVI-B from the lunar transit. *Top row:* Map of $u_i^\star/f_i$ ratios on the lunar disk in one of the 8 transit images. *Bottom row:* Histogram of the vector R_B containing the values of the ratios $u_i^\star/f_i$ on each of the eight lunar disks. The bar height represents the fraction of pixels within 0.015 of a given $u_i^\star/f_i$ ratio value.

	68^{th}	95^{th}	99.7^{th}
171 Å	0.06	0.11	0.16
195 Å	0.05	0.10	0.16
284 Å	0.07	0.14	0.23
304 Å	0.04	0.09	0.14

Table 4.3: The 68^{th}, 95^{th}, and 99.7^{th} percentile values of the relative error vector $|R_B|$ for each filter band of EUVI-B.

The histogram of R_B is shown in the bottom row of Fig. 4.8. The histograms for 171, 195, and 304 Å all have similar shapes, but the 284 Å histogram is somewhat wider, indicating a tendency towards larger errors. The wider 284 Å histogram is due to the uncorrected shadowing by the backup filter in 284 Å, and the artifacts caused by h^g as discussed above. We measure the spread of R_B using percentiles of $|R_B|$, which are reported in Table 4.3. We report the 68^{th}, 95^{th}, and 99.7^{th} percentiles, which correspond to the first three standard deviations of a Gaussian. Note that the values of $|R_B|$ at these percentiles increase by a roughly constant amount with each standard deviation, as is expected of a Gaussian error. In 171 Å, for example, the percentile values are 0.06, 0.11, and 0.16, increasing by about 0.05 with each standard deviation. This suggests that ϵ_{psf}/f can be reasonably modeled as a Gaussian with standard deviation given by the 68^{th} percentile.

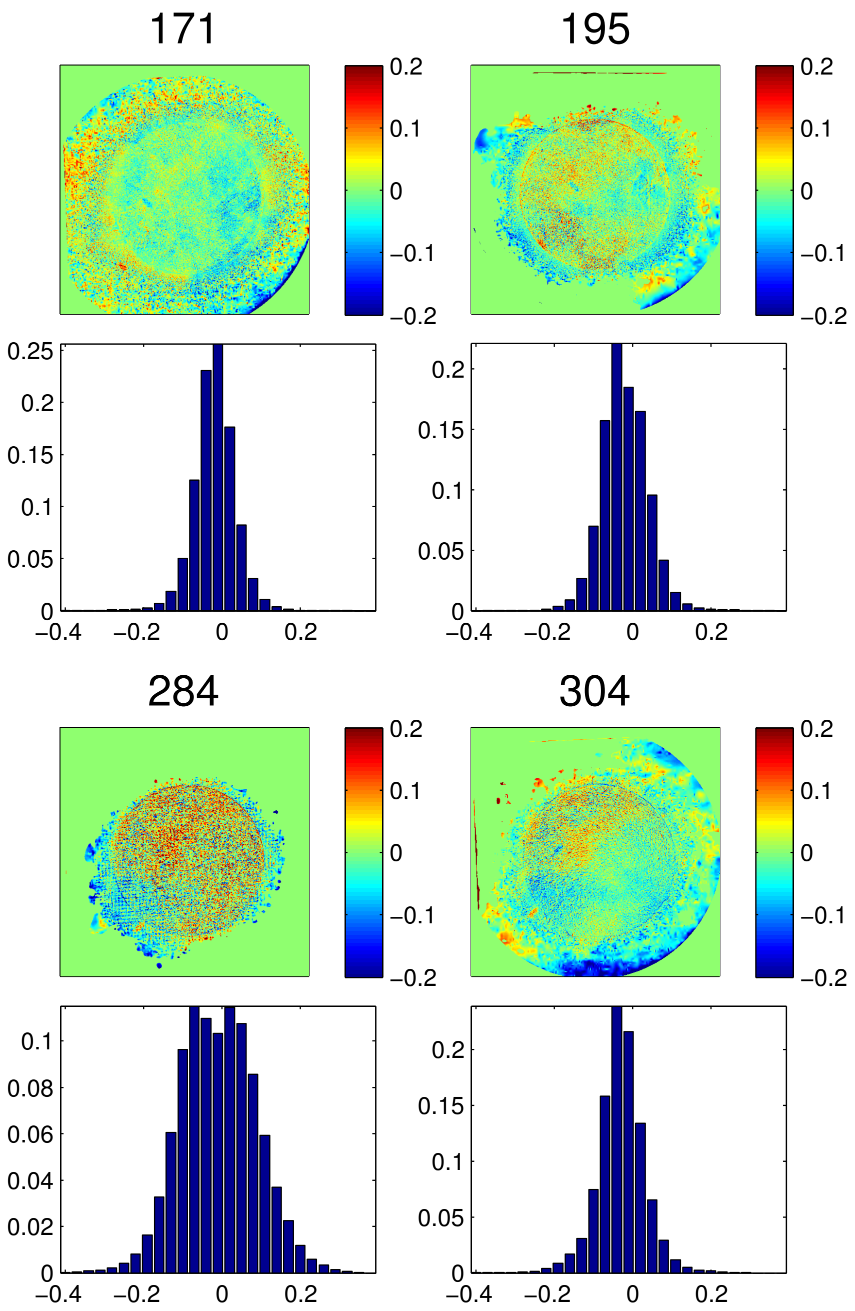

Figure 4.9: Relative discrepancy between simultaneous exposure EUVI-A and B images from the early mission. For 171, 284, and 304 Å the images are taken on December 22, 2006 from 01:00:56 to 01:01:28 UTC, while for 195 Å they were taken on December 14 at 18:45:00 UTC. *Rows 1 and 3:* The relative difference map $(u_A - u_B)/f_A$. *Rows 2 and 4:* Histograms of $(u_A - u_B)/f_A$.

	68^{th}	95^{th}	99.7^{th}
171 Å	0.02	0.08	0.16
195 Å	0.02	0.09	0.20
284 Å	0.05	0.17	0.28
304 Å	0.01	0.08	0.20

Table 4.4: The 68^{th}, 95^{th}, and 99.7^{th} percentile values of $|(u_A - u_B)/f_B|$ for each EUVI filter band.

	68^{th}	95^{th}	99.7^{th}
171 Å	0.07	0.12	0.23
195 Å	0.07	0.12	0.22
284 Å	0.12	0.20	0.32
304 Å	0.07	0.13	0.24

Table 4.5: The 68^{th}, 95^{th}, and 99.7^{th} percentile values of the relative error map $|R_A|$ for each filter band of EUVI-A.

4.4.3 Estimating ϵ_{psf} in EUVI-A

Since EUVI-A has not observed a transit, we lack information about the ideal EUVI-A image u_A^{true} and cannot directly estimate error in the deconvolved EUVI-A images u_A. However, we can obtain an indirect estimate from early mission images: the ideal image u^{true} is the same for EUVI-A and B, so the corrected EUVI-B image u_B provides an estimate of u^{true}.

Formally, suppose we are given two simultaneous early mission observations f_A and f_B. We have from (4.31) that

$$\begin{aligned} u_A &= u^{\text{true}} + \epsilon_{\text{psf},A} + \epsilon_{\text{bv},A} \\ u_B &= u^{\text{true}} + \epsilon_{\text{psf},B} + \epsilon_{\text{bv},B}. \end{aligned} \tag{4.35}$$

Subtracting u_B from u_A gives

$$u_A - u_B = (\epsilon_{\text{psf},A} - \epsilon_{\text{psf},B}) + (\epsilon_{\text{bv},A} - \epsilon_{\text{bv},B}), \tag{4.36}$$

a sum of errors due to PSF inaccuracy and bias variation. As in the EUVI-B analysis, we must identify and ignore those pixels where $\epsilon_{\text{psf},A} - \epsilon_{\text{psf},B}$ cannot be safely assumed to dominate $\epsilon_{\text{bv},A} - \epsilon_{\text{bv},B}$. The estimated magnitude of $\epsilon_{\text{bv},A} - \epsilon_{\text{bv},B}$, which we call $\Delta\sigma_{\text{bv}}$, is calculated by adapting (4.48) to estimate the bias variation in a difference of two images:

$$\Delta\sigma_{\text{bv}}^2 = \frac{1}{4-1}\sum_{i=1}^{4}((c_{i,A} - c_{i,B}) - (\overline{c}_A - \overline{c}_B))^2, \tag{4.37}$$

where $c_{i,A}$ and $c_{i,B}$ are the means of the vignetted corners of f_A and f_B, and $\overline{c}_A = \frac{1}{4}\sum_{i=1}^{4} c_{i,A}$ and $\overline{c}_B = \frac{1}{4}\sum_{i=1}^{4} c_{i,B}$ are the grand means. We ignore the pixels x where $|f_A(x) - f_B(x)| \leq 2N_b\Delta\sigma_{\rm bv}$, and outside of these pixels we assume $u_A - u_B$ is roughly $\epsilon_{{\rm psf},A} - \epsilon_{{\rm psf},B}$ and write $\epsilon_{{\rm psf},A}$ as the sum of two errors:

$$\epsilon_{{\rm psf},A} = (u_A - u_B) + \epsilon_{{\rm psf},B}. \tag{4.38}$$

We then calculate an estimate $\tilde{\sigma}^2_{{\rm psf},A}$ of $\epsilon_{{\rm psf},A}$ by adding $u_A - u_B$ in quadrature with the previously calculated estimate $\sigma_{{\rm psf},B} = \rho_B f_B$ of PSF error in u_B:

$$\tilde{\sigma}^2_{{\rm psf},A} = (u_A - u_B)^2 + (\rho_B f_B)^2, \tag{4.39}$$

where ρ_B is a constant measuring the spread of $\epsilon_{{\rm psf},B}/f_B$ determined in the previous section. We set ρ_B to the 68^{th} percentile in Table 4.3.

Our estimates of $\epsilon_{{\rm psf},A}/f_A$ are derived from the distribution of $R_A = \tilde{\sigma}_{{\rm psf},A}/f_A$, which is treated as a surrogate for the more direct error measurements R_B provided by the lunar transit. Percentiles of R_A are used to obtain a value ρ_A analogous to ρ_B.

To guard against overfitting, we apply the error analysis to early mission images as dissimilar as possible from the images used to determine the PSF. For 171, 284, and 304 Å we used images from simultaneous A/B exposures on December 22, 2006 from 01:00:56 to 01:01:28 UTC, almost a week after the exposures used to fit the PSF. This series did not include 195 Å so in its case we used a simultaneous exposure from December 14 at 18:45:00 UTC, one hour after the exposure used to find the PSF. The ratio $(u_A - u_B)/f_A$ is shown for each band in Fig. 4.9, *top row.* Choosing ρ_B as the 68^{th} percentile from Table 4.3, we computed for R_A the percentiles listed in Table 4.5.

4.5 Results

The contribution of stray light is generally greatest in faint regions that appear near bright regions. Thus areas off the limb, coronal holes, filaments, and filament cavities are expected to receive the largest downward revisions when deconvolution is applied. Here we present EUVI images of such features before deconvolution (f) and after (u). The ratio map u/f is presented to illustrate where the correction is large, and line plots with error bars are given. Error bars σ_u for the deconvolved image u are computed using (4.32), and the component terms are computed as described in §4.4. The values of ρ_A and ρ_B used to compute $\sigma_{\rm psf}$ are set to the 95^{th} percentile in

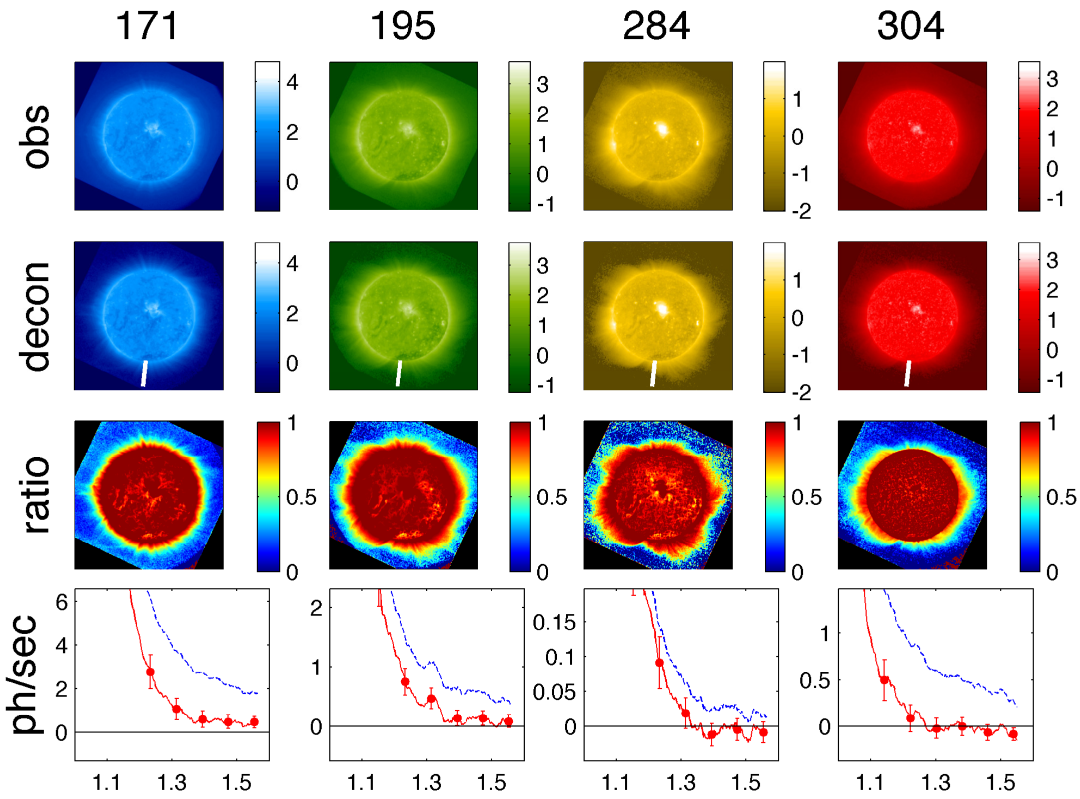

Figure 4.10: The global corona as seen by EUVI-A on Jan. 19, 2007 (UTC 01:00:56 to 01:01:28). A 16 × 16 moving average has been applied for noise suppression. *Rows 1 and 2:* The observed data f and corrected data u ($\log_{10}$ ph/sec). A white line has been drawn between the plumes below the South pole. *Row 3:* The ratio image u/f. *Row 4:* Intensities of f (dotted line) and u (solid line) along the white line segment as a function of distance from solar center (units of $R_\odot$). Error bars are given every 50 pixels.

Tables 4.5 and 4.3 respectively.

In Figs. 4.10 and 4.11 we examine the global corona as seen by EUVI-A and B on January 19, 2007, focusing on the intensity profile versus height above the limb. The exposures were not ICER compressed and thus provide the most reliable off-limb intensity information. Near the limb deconvolution generally has little effect, but as we move away from the limb the ratio u/f rapidly drops to less than 50%, with many regions dropping to 10% or lower. This implies that at least 50% of the apparent off-limb intensity is stray light, and in some areas it may rise to 90% or more. We plot the observed and deconvolved image intensity on a line above the North pole, and find that the intensities decay much faster in the deconvolved images.

Figs. 4.12 and 4.13 shows a coronal hole seen by EUVI-A on October 3, 2008 and

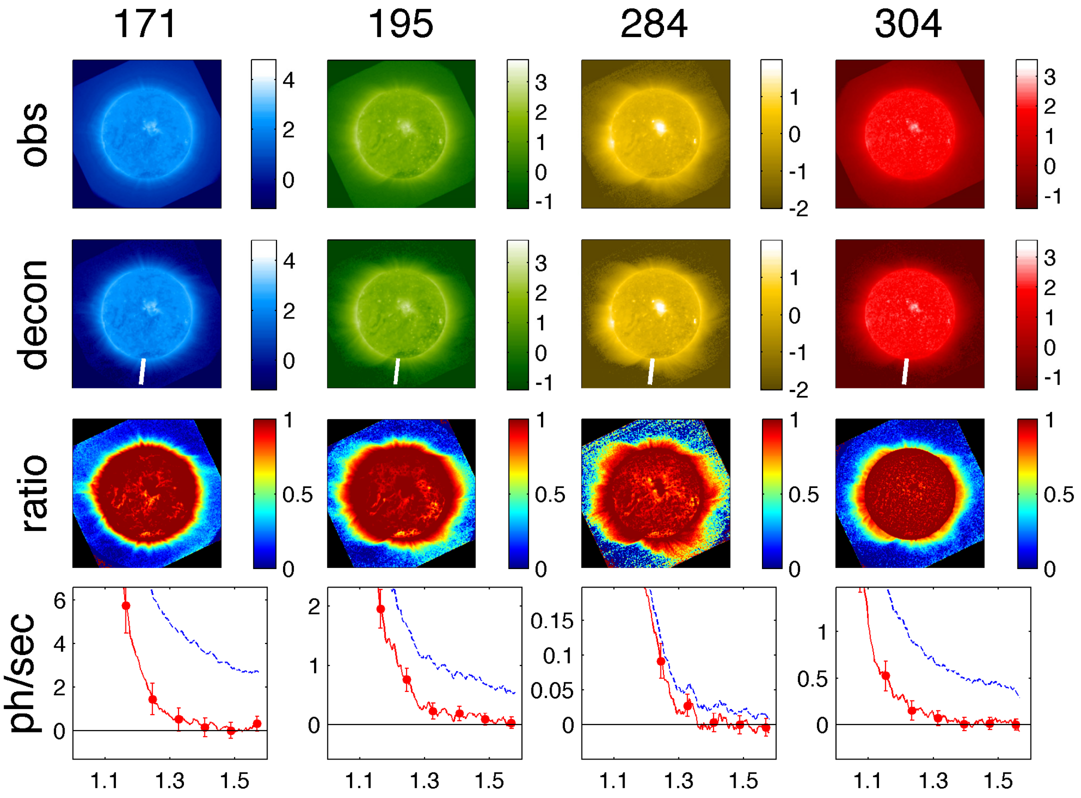

Figure 4.11: The global corona as seen by EUVI-B on Jan. 19, 2007 (UTC 01:00:56 to 01:01:28). A 16 × 16 moving average has been applied for noise suppression. *Rows 1 and 2:* The observed data f and corrected data u ($\log_{10}$ ph/sec). A white line has been drawn between the plumes below the South pole. *Row 3:* The ratio image u/f. *Row 4:* Intensities in ph/sec of f (dotted line) and u (solid line) along the white line segment as a function of distance from solar center (units of $R_\odot$). Error bars are given every 50 pixels.

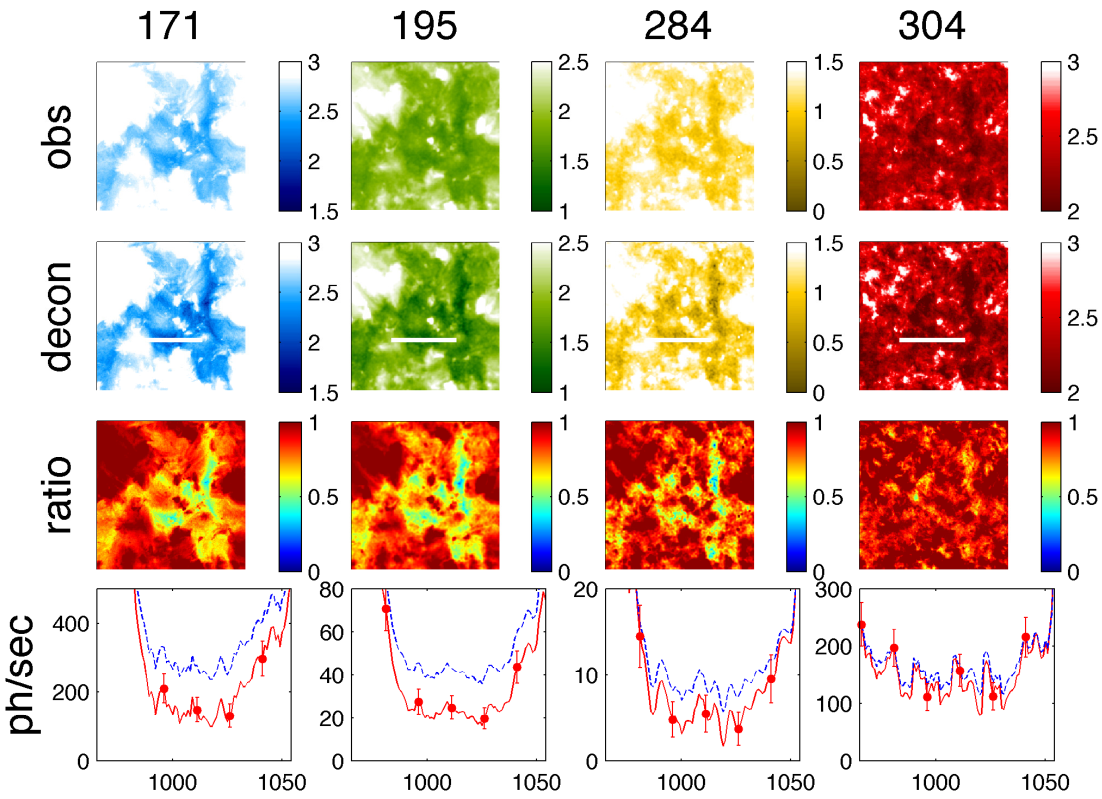

Figure 4.12: Stray light correction of an on-disk coronal hole observed by EUVI-A on Nov. 20, 2008. A 3 × 3 moving average has been applied to the 195 and 284 Å images for noise suppression. *Rows 1 and 2:* The observed data f and corrected data u ($\log_{10}$ ph/sec). A white line has been drawn over the middle of the coronal hole. *Row 3:* The ratio image u/f. *Row 4:* Intensities in ph/sec of f (dotted line) and u (solid line) versus image pixel index along the white line segment. Error bars are given every 20 pixels.

EUVI-B on November 21, 2008. In all bands except 304 Å, the stray light corrected coronal holes are significantly dimmer: the ratio u/f ranges from 30% to 60% in the deepest parts of the coronal holes, implying that 40-70% of the apparent coronal hole emissions are stray light.

The effect of stray light correction can be highly dependent on the filter band. To illustrate this, a filament cavity observed on the limb by EUVI-B is shown in Fig. 4.14. The ratio images show that up to 60% of the cavity is stray light in 171 and 304 Å, but in 195 and 284 Å the stray light levels are much lower. Stray light levels in a given region depend on the PSF and the brightness of neighboring features, both of which vary considerably with the region's location and the band.

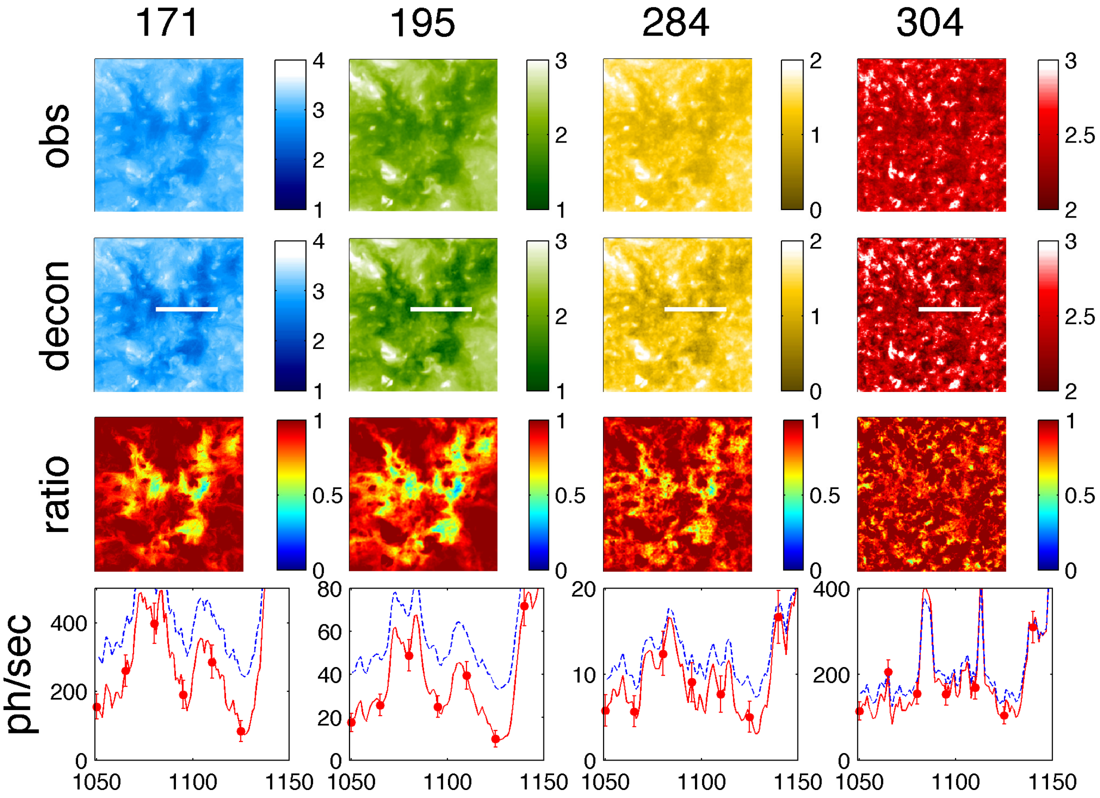

Figure 4.13: Stray light correction of an on-disk coronal hole observed by EUVI-B on Nov. 20, 2008. A 3 × 3 moving average has been applied to the 195 and 284 Å images for noise suppression. *Rows 1 and 2:* The observed data f and corrected data u ($\log_{10}$ ph/sec). A white line has been drawn over the middle of the coronal hole. *Row 3:* The ratio image u/f. *Row 4:* Intensities in ph/sec of f (dotted line) and u (solid line) versus image pixel index along the white line segment (units of ph/sec). Error bars are given every 20 pixels.

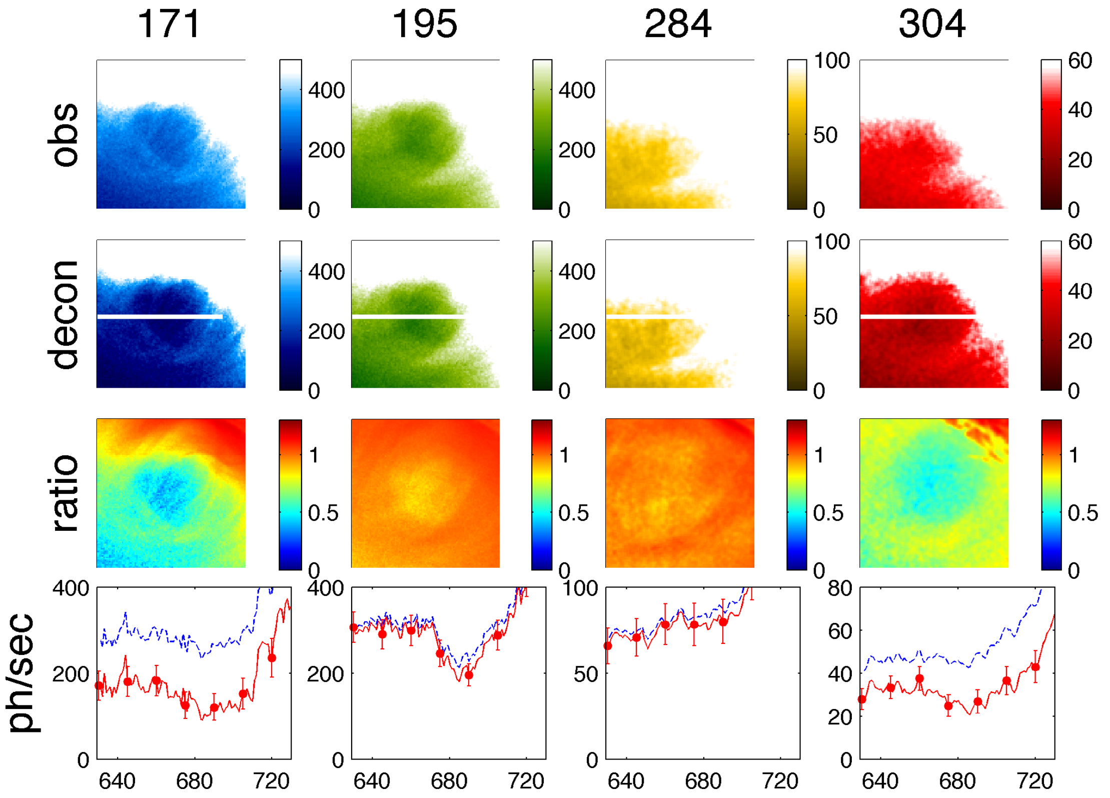

Figure 4.14: Stray light correction of a filament cavity observed by EUVI-B on Feb. 25, 2007. A 3 × 3 moving average has been applied to the 284 and 304 Å images for noise suppression. *Rows 1 and 2:* The observed data f and corrected data u (ph/sec). A white line has been drawn over the middle of the filament cavity. *Row 3:* The ratio image u/f. *Row 4:* Intensities of f (dotted line) and u (solid line) versus image pixel index along the white line segment. Error bars are given every 15 pixels.

4.6 Conclusion

We have determined PSFs for all filter bands of the EUVI instruments aboard STEREO-A and B via a maximum likelihood-type method. The method is based on a model of EUV image formation that accounts for long-range scatter, photon noise, and CCD noise. Scatter was assumed to arise from a spatially invariant PSF h, which was represented as the convolution of a mesh component h^g and an empirical component h^m. The mesh component represents diffraction by the mesh over the entrance aperture, while the empirical component represents scattering by mirror microroughness and other effects. It was shown that if the effects of Fresnel diffraction and the pupil are neglected, this model can be derived from a Fourier optics model of EUVI. Whether these effects are in fact negligible is unclear, and we suspect that our PSFs could be improved substantially with a more complete instrument model.

The mesh component was determined using Fraunhofer diffraction theory. The empirical component was modeled using a generalized Lorentzian formula found in the EUV mirror literature, but this model required some modification to obtain an empirically workable stray light correction. Accounting for the stray light anisotropy was found to be essential for a high quality correction, particularly off the limb. Anisotropy was accounted for by allowing h^m to have an elliptical shape with an orientation determined by analysis of calibration roll data.

A consistent correlation was observed between the direction of anisotropic scatter and the scatter predicted from h^p, the PSF resulting from Fraunhofer diffraction by an ideal quarter annulus pupil. A similar correlation was observed in SOHO/EIT [5]. We showed that the long-range wings of h^p do not contain enough mass to explain the observed stray light distribution, but it is possible that h^p underestimates scatter due to the pupil: for example, Fresnel diffraction effects between the optical elements or a non-ideal pupil with rough edges could make the wings of h^p heavier. These possibilities can and should be explored quantitatively via computational models developed in collaboration with optical engineers.

We determined the PSF for each EUVI-B band by applying semiblind deconvolution to lunar transit data, which solves for the stray light corrected image and PSF simultaneously. Thanks to a variable elimination technique, our method is no more computationally demanding than previous methods, which rely on a predetermined heuristic approximation of the corrected image to obtain the PSF. For EUVI-A we exploited the fact that A and B observed the Sun from the same position in December 2006. We deconvolved the B image with its estimated PSF to approximate the corrected image, which enabled determination of the A PSFs.

The determined PSFs enable quick correction of any EUVI image for stray light, and our error analysis enables quick estimates of both systematic and random error to be calculated for any deconvolved image. While we have considered only uncompressed images in this paper, stray light correction can also be usefully applied to ICER compressed images, which comprise the vast majority of the EUVI database. Care must be taken with these images as they have more artifacts and degraded accuracy, particularly off the limb. The suitability of stray light correction for ICER images will be addressed in an upcoming publication.

We have shown the effects of stray light correction on various solar features. The large downward corrections to faint regions have major impacts on the plasma diagnostics available from EUV images. A downward correction of the observed intensity causes a proportional reduction of the estimated value of n_e^2 and analysis of the 171, 195 and 284 Å intensities show a downward revision of the coronal hole column density $[\int_{\mathrm{LOS}} \mathrm{d}l \, n_e^2(l)]^{1/2}$ of up to 40%. The effect on the estimated T_e is more complex. The removal of stray light from the off-limb causes a dramatic steepening of the profile function $n_e(h)$ (where h is the height above the photosphere). The specific impact of stray light correction, including constraints on solar wind models from the corrected profiles $n_e(h)$, $T_e(h)$, is currently under investigation. Obvious consequences include reduction of the plasma β, electron-ion collision rates, and the mass of solar wind plasma requiring acceleration.

Similar methods may be applied to treat the stray light problems in the other solar EUV imaging instruments (SOHO/EIT, TRACE, STEREO-A/EUVI, SDO/AIA). This work and its heliophysical implications will be reported in more detail in upcoming publications.

4.7 Appendix

4.7.1 Optical modeling

Scalar diffraction theory can be used to mathematically model scattering in EUVI and determine a PSF. In Fig. 4.15, we propose a model in which the mirrors are represented by thin lenses and random phase screens, and the wire meshes and aperture masks are represented by amplitude screens. The model consists of four planes representing the entrance aperture, primary mirror, secondary mirror, and CCD respectively. The incident field propagates between these planes by Fresnel diffraction. When the field encounters a plane, it is perturbed by any screens or lenses present in

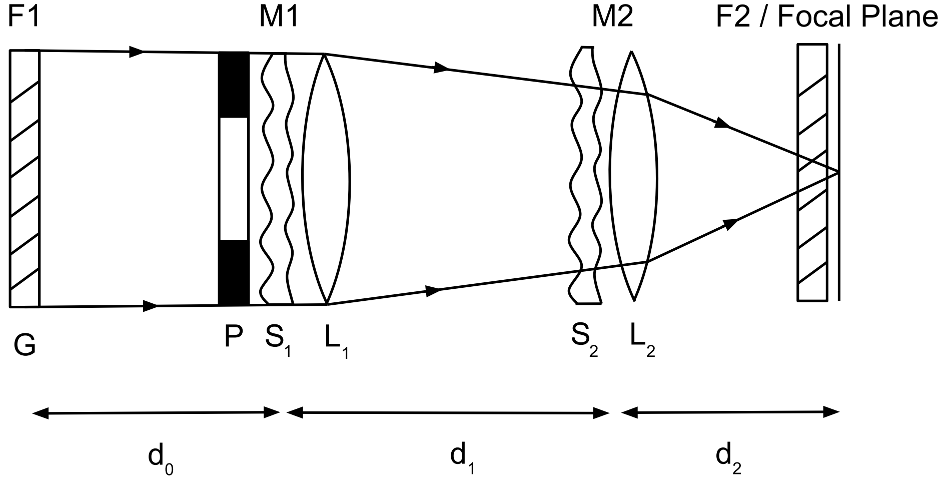

Figure 4.15: Schematic diagram illustrating the mathematical model of the EUVI optical system expressed in (4.41). The instrument is treated a series of four planes - entrance aperture plane, primary mirror plane, secondary mirror plane, and focal plane - separated by distances d_0, d_1, and d_2. The entrance aperture plane contains an amplitude screen G representing the wire mesh at F1. The primary mirror plane contains an amplitude screen P representing the quarter annulus pupil, a random phase screen S_1 representing mirror microroughness, and a thin lens L_1 representing the phase shift imposed by an ideal mirror surface. The secondary mirror plane is similar but the amplitude screen is omitted for simplicity. The backup rejection filter F2 in front of the focal plane is pictured here but omitted from the PSF modeling because it causes shadowing and not scatter.

that plane. The final plane is the focal plane. The backup rejection filter F2 near the focal plane does not scatter light, but only shadows the CCD, so it is not considered in the PSF determination. For simplicity we ignore the finite extent of the aperture in all planes except the primary mirror plane, which contains a quarter annulus screen representing the aperture mask in front of the primary mirror [55]. The finite extent of the entrance aperture and secondary mirror could be represented by adding quarter annulus amplitude screens to their planes.

The electric field in the focal plane can be expressed compactly by applying a series of linear operators to the field at the entrance aperture plane. This idea is presented by Goodman in [49, Chapter 5.4], and a variant of his operator notation is adopted here. We let $\mathcal{M}_V[U](w) = V(w)U(w)$ denote pointwise multiplication of the input field $U(w)$ by the screen $V(w)$, where $V(w)$ may be complex-valued and include both phase and amplitude components. The operation of Fresnel propagation of U over a distance d between two parallel planes is given by

$$\mathcal{R}_d[U](w) = \frac{e^{ikd}}{i\lambda d} \int U(w')e^{i\frac{k}{2d}|w'-w|^2} dw', \tag{4.40}$$

where $k = 2\pi/\lambda$, $|w| = \sqrt{\langle w, w\rangle}$ is the length of the vector w in the plane $\mathbb{R}^2$, and the integration is over the plane [49, Chapter 4.2]. We give the pupil plane and focal plane physical coordinates $\xi \in \mathbb{R}^2$ and $X \in \mathbb{R}^2$ aligned with the CCD primary axes, with origins determined by intersecting each plane with the path of a normally incident geometric ray. Letting $U_p(\xi)$ denote the pupil plane field, $U_f(X)$ the focal plane field, we have

$$U_f = (\mathcal{R}_{d_2} \cdot \mathcal{M}_{S_2L_2} \cdot \mathcal{R}_{d_1} \cdot \mathcal{M}_{PS_1L_1} \cdot \mathcal{R}_{d_0} \cdot \mathcal{M}_G)[U_p], \tag{4.41}$$

where the dot denotes composition of operators. Radiation from the solar surface is spatially incoherent, so the focal plane PSF is proportional to the field intensity $|U_f|^2$ produced by a plane wave U_p [49, Chapter 6]. This model is expected to generate a PSF that varies with position in the focal plane, which would present serious computational challenges. Even if this variation is negligible, as we have assumed in this paper, the presence of multiple Fresnel integrals makes the PSF difficult to compute. However, the fractional Fourier transform can be used to approximate Fresnel integrals numerically [85], which may enable future work within this model.

In the present work, we obtain a simpler model (at the expense of physical accuracy) by removing $\mathcal{R}_{d_0}$ and P, then moving the remaining amplitude and phase screens G, S_1, S_2 to the plane containing the first lens:

$$U_f \approx (\mathcal{R}_{d_2} \cdot \mathcal{M}_{L_2} \cdot \mathcal{R}_{d_1} \cdot \mathcal{M}_{L_1}) \cdot \mathcal{M}_{GS_1S_2}[U_p]. \tag{4.42}$$

The term in the parentheses represents an optical system free of scattering by diffraction and mirror microroughness. The system is still subject to geometric aberrations, but these are significant only on a scale of 1-2 pixels (J.-P. Wuelser, *personal communication*). Above this scale the system may be considered geometrically ideal, and the term in parentheses may be represented as a scaled Fourier transform:

$$U_f \approx (\mathcal{S}_{1/Z\lambda} \cdot \mathscr{F}) \cdot \mathcal{M}_{GS_1S_2}[U_p], \tag{4.43}$$

where $\mathcal{S}_{1/Z\lambda}[U](X) = U(X/Z\lambda)$ and $Z = 1750$ mm is the EUVI effective focal length. Setting $U_p(\xi)$ to be a normally incident plane wave ($U(\xi) = 1$ for all ξ) and setting $S = S_1S_2$, we obtain a formula for the instrument PSF h_0 in physical units X and pixel units x:

$$h_0 \propto |U_f(X/Z\lambda)|^2 \approx |\mathscr{F}[GS](X/Z\lambda)|^2 = |\mathscr{F}[GS](\tau x)|^2. \tag{4.44}$$

In the second line we have used the plate scale relation $\Delta\theta_p x = (X/Z)$ and the constant $\tau = \Delta\theta_p/\lambda$.

The screen $S(\xi) \propto \exp(-2\pi i\phi(\xi))$ represents the net phase error accumulated by the incoming wavefronts after reflection off the primary and secondary mirrors. The phase function $\phi(\xi)$ is far too complex to be modeled deterministically, so it is typically treated as a Gaussian random field. Under this model there is no simple formula for the specific PSF h_0 realized in the instrument, but the expected PSF $h = \langle h_0 \rangle$, averaged over all realizations of the random field $\phi(\xi)$, has a simple form which we adopt as our PSF model. In [50, Chapter 8.1] a general calculation applicable to this model gives

$$h = \langle h_0 \rangle = h^g * h^m, \tag{4.45}$$

where $h^g(x) \propto |\hat{G}(\tau x)|^2$ and $h^m \propto \langle |\hat{S}(\tau x)|^2 \rangle$ will be called the grid and the empirical PSF respectively. The two proportionality constants are determined by the constraint that the PSF must integrate to unity (or sum to unity in the discrete case).

It is unclear how much error is incurred by the heuristic simplifications used to obtain this model. Even if it is justifiable in some sense to move the screens, they may need to be modified to compensate for neglecting the effects of Fresnel propagation. It seems likely that the effects we have neglected contribute to the empirical anisotropy of the PSF and our deconvolution's systematic error.

4.7.2 Image preparation

The purpose of this appendix is to describe the known effects in a raw EUVI image, f_{raw}, and the steps required to obtain an image that conforms to this model well enough for the purposes of PSF determination and error analysis. We begin by recalling that, before shadowing by the mesh F2 supporting the backup rejection filter, the expected image in the focal plane is $h^{\text{true}} * u^{\text{true}}$. Letting $F(x)$ denote the fraction of light admitted by the mesh at pixel x, the expected image after F2 is $F(h^{\text{true}} * u^{\text{true}})$, and the number of photons actually captured in an exposure is $f_{\text{phot}} = F(h^{\text{true}} * u^{\text{true}}) + n_{\text{phot}}$, and the digital number (DN) readout of the detector is γf_{phot}, where γ is the photoelectric gain constant (DN/photon). The raw images also have additive CCD noise n_{ccd}, impulse noise n_{spike} due to cosmic rays, and bias B, so the raw image in DN is given by

$$f_{\text{raw}} = \gamma f_{\text{phot}} + n_{\text{ccd}} + n_{\text{spike}} + B. \tag{4.46}$$

The prepared image f is obtained by removing the bias, despiking, and dividing by γF. Assuming these corrections are accurate we have

$$f = \frac{1}{\gamma F}(f_{\text{raw}} - B - n_{\text{spike}}), \tag{4.47}$$

which leads to (4.11), the model used to determine the PSFs.

The EUVI image preparation procedure `euvi_prep.pro` used in SolarSoft is responsible for removing B, while the despiking utility `despike_gen.pro` is used to remove n_{spike}. For most purposes these corrections are entirely satisfactory. However, the demands of stray light correction are unusual, particularly in the highly informative but very faint regions far off the limb. Here we describe a custom procedure for estimating and removing bias and spikes. We also show that the bias exhibits large-scale systematic variation, meaning that B is not constant but varies slowly with position in the image. This variation is estimated and included in our analysis of error in deconvolved images.

4.7.2.1 Bias estimation and removal

The `euvi_prep.pro` debiasing procedure corrects CCD bias, which accounts for almost all of the bias B. The CCD bias estimate it subtracts is obtained from the

FITS header (keyword `BIASMEAN`). The `BIASMEAN` estimate is obtained by sampling values from the *overscan*, a collection of pixel values read from the CCD electronics without collecting any electrons from the physical CCD pixels. These measure the contribution of the on-chip amplifier to the readout [58]. Depending on the image, up to 128 overscan rows and 64 overscan columns are collected, enlarging the image from 2048 × 2048 to 2176 × 2112. The FITS header estimate for B uses only one column of the overscan, and in some images we have observed this column's mean to differ from the mean value over the whole overscan by up to 0.5 DN. This is enough to complicate our error analysis, so we initially estimate B by taking the mean of the whole overscan instead of using the FITS header value.

Even after this constant is removed, the mean values of the rows of the overscan exhibit variability. This is typically between 0.2 and 0.5 DN peak-to-peak, and occurs mostly in first few dozen rows. Assuming that the variability is constant along each row (i.e. it does not change with column index), the variability can be somewhat compensated by forming a vector v of the mean values of each overscan row, then subtracting v_i from the i^{th} row of the image. To reduce noise, v is smoothed by applying a moving average of length 20.

Initial difficulties in getting EUVI-A and B deconvolutions to agree off the limb led us to suspect that there may be contributions to the bias beyond what is represented in the overscan. To test for this we examined vignetted pixels near the four corners of each EUVI image. Vignetting occurs because there is a circular filter wheel in front of the CCD, and the CCD diagonal is slightly longer than the filter wheel diameter. We visually identified the boundary between vignetted and unvignetted pixels in each EUVI-A and B image, confirming that it is composed of four circular arcs formed by the intersection of the square CCD array with the circular filter wheel shadow. We then selected about a dozen pixels deep in the vignetted region for each image and fit a circle to these pixels. The pixels outside the fitted circle formed four disjoint sets, one for each corner of the square, and were assumed to be fully vignetted.

We computed the mean readouts $c_1, \ldots, c_4$ of each corner, the grand mean $\overline{c} = \frac{1}{4}\sum_{i=1}^{4} c_i$, and the unbiased variance estimate

$$\sigma_{\mathrm{vc}}^2 = \frac{1}{4-1}\sum_{i=1}^{4}(c_i - \overline{c})^2 \tag{4.48}$$

The quantity σ_{vc} represents large-scale variations in the bias which cause the four corner means to differ. In Fig. 4.16 the values of $\overline{c}$ and σ_{vc} are shown for the 8

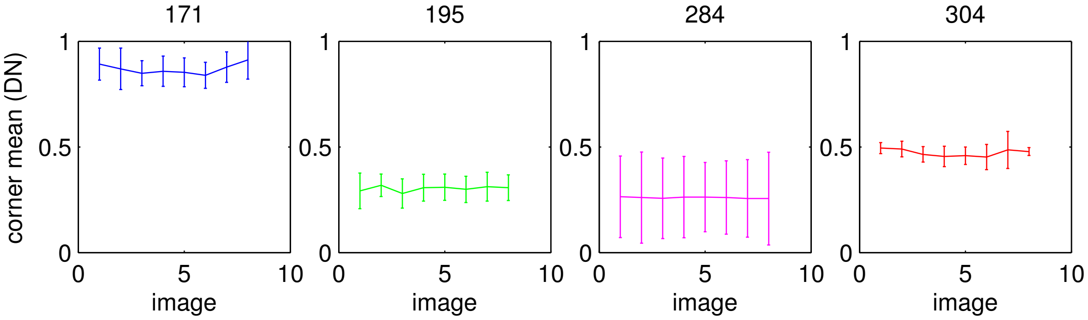

Figure 4.16: Grand mean $\bar{c}$ and RMS deviation from the grand mean $\sigma_{\rm vc}$ for the vignetted corner means $c_1, \ldots, c_4$ over the 8 lunar transit images in each EUVI band.

uncompressed images (in each band) from the Feb. 25, 2007 lunar transit. The average values of $\bar{c}$ for each wavelength are 0.9, 0.3, 0.25, and 0.5 DN for 171, 195, 284, and 304 Å respectively.

The computed values of $\bar{c}$ and $\sigma_{\bar{c}}$ are far too large and regular to be attributed to normal statistical variability in $n_{\rm ccd}$. Each of the selected corner regions contain over 16,000 pixels and each corner set has a total of at least 160,000 pixels, so the expected standard deviation of $\bar{c}$ due to CCD noise is $\sqrt{\langle\sigma_{\bar{c}}^2\rangle} \leq \sigma_{\rm ccd}/\sqrt{160000} = 0.0025$ DN. The computed values of $\bar{c}$ are highly stable and much larger than this value. The expected value of $\sigma_{\rm vc}$ can be estimated by adding in quadrature the standard errors in the means of the four corners: $\langle\sigma_{\rm vc}^2\rangle = \sum_{i=1}^4 \sigma_{c_i}^2 = \sum_{i=1}^4 \sigma_{\rm ccd}^2/n_i$, where n_i is the number of pixels in each corner. From this we compute a value of $\langle\sigma_{\rm vc}^2\rangle^{1/2} = 0.006$ DN in each band. The computed values of $\bar{c}$ and $\sigma_{\rm vc}$ are quite stable and much larger than the expectations, so they must be due to some additional bias beyond that measured by overscans.

Dark current undoubtedly contributes some of this bias, but cannot fully explain some features of it. For example, dark current is a function of exposure time and detector temperature, but not of wavelength, since it is not generated by EUV photons. But the two bands 171 and 195 Å both used 20 second exposures during the transit and have $\bar{c}$ values around 0.9 and 0.3 DN respectively. Thus the cause of the unexpectedly high $\bar{c}$ and $\sigma_{\rm vc}$ values is unclear, and may involve subtle CCD behavior, low-level stray light leaks, or scattering by the backup filter mesh F2. Whatever the cause, we subtract the constant $\bar{c}$ from the image as part of our preparation.

4.7.2.2 Despiking

The SolarSoft utility `despike_gen.pro` is normally used for despiking EUVI images. It is works well for many applications, but has difficulty removing spikes that span many pixels. Uncorrected cosmic ray impacts off the limb can create errors hundreds of sigmas above the nominal noise level from §4.3, which are large enough to degrade the PSF parameter estimates.

To eliminate these we augment the standard procedure with a rather aggressive second step, which is applied only to the faintest areas off the solar limb where there is minimal structure capable of causing false positives. For each image to be processed, we first formed a mask by identifying the fraction $p = 1 - \pi(f_s R_\odot)^2/N_f$ of pixels with lowest intensity, where N_f is the number of unvignetted pixels, $R_\odot$ is the solar radius in pixels, and $f_s = 1.15, 1.05, 1.03, 1.15$ in 171, 195, 284, and 304 Å respectively. On-disk pixels were excluded from the mask. The mask was then enlarged to fill in 'gaps' caused by noise: any pixel within a box of side-length 32 pixels around a mask pixel was added to the mask. Empirically, we found that the resulting mask was composed of the pixels in areas far off the limb, where there is minimal small-scale structure. We then calculated a 11×11 median filtered image and identified any pixels that were in the mask and more than $3\sigma_f$ above the median as spikes. Here, σ_f denotes the estimate of photon and CCD noise in the original image defined in §4.3. The identified spikes were replaced with values from the median filtered image. After careful examination of the images used in PSF fitting and error analysis, we concluded that this procedure removed almost all spikes, and no correlation of identified spikes with solar structure was observed.

4.7.3 Wavelet denoising

Here we describe the procedure used to reduce noise in the EUVI images that form the basis of the systematic error analysis of §4.4. In the images we analyze, noise is generally significant only in very faint areas off the limb. In these areas, however, the images tend to be fairly noisy even after the 4×4 binning applied before error analysis. To smooth off-limb noise away without disturbing solar structure, we performed wavelet denoising on all analyzed images.

The wavelet transform uses a pair of filters to decompose an image into a low-pass component containing large scale structure and a high-pass component containing small-scale details [75], and can be inverted by filtering and summing these components together. In the standard transform, both components are subsampled to maintain the original signal dimension, but there is also a translation invariant version which omits the subsampling. The decomposition step is repeated $J-1$ times on the low-pass component to obtain a multi-scale transform having $J+1$ components: J detail components containing local fluctuations at progressively larger scales, and a final low-pass component containing the global behavior. *Lifting schemes* are used for the technical implementation of more recent wavelet transforms [100].

Wavelet high-pass filters are generally designed to have *vanishing moments* of some order d, meaning (roughly) that the filter has no response to a polynomial trend of degree $d-1$ or less. In general, the decomposition and reconstruction filters have different vanishing moments d_1 and d_2, and a wavelet transform with such vanishing moments is called a d_1/d_2 transform. Vanishing moments help to ensure that large-scale signal trends are held in the low-pass coefficients alone. For example, if a wavelet transform is applied to a signal composed of a linear trend and additive noise, the detail components receive most of the noise and almost none of the linear trend, while the final low-pass component holds a denoised version of the linear trend.

Wavelet denoising works by applying a wavelet transform to the image, setting small detail transform coefficients to zero, then transforming back. We used a translation invariant 5/3 lifting transform with $J=5$ levels and applied thresholding only to detail coefficients below a threshold of $2\sigma_{\text{ccd}}$. Denoising generally does not affect features on-disk and near the limb because their wavelet coefficients are too large. Off the limb, noise is removed without disturbing the large scale structure, which is contained almost entirely in the low-pass coefficients.

4.7.4 Calibration rolls and stray light anisotropy

We examine STEREO calibration roll images to reveal the anisotropic stray light distribution and test our deconvolution's ability to correct it. On November 29, 2011, STEREO-A executed a 360° calibration roll, and in each band it acquired 9 solar images at roll angles of $\theta = 0$, 60, 90, 120, 180, 240, 270, 300, and 360° relative to the pre-roll position. STEREO-B performed an identical roll on November 8, 2011. Images for each band were acquired at a cadence of 20 minutes. Since the PSF is determined by the instrument optics, it rotates with the instrument, so the distribution of stray light rotates with respect to the pre-roll image coordinate system.

To track this rotation we coaligned all images to the coordinates of the pre-roll image, giving a series of 9 images $f_0, f_{90}, \ldots, f_{360}$ for each band, and examined the difference images $\Delta f_\theta = f_\theta - f_0$. The 90° difference image $\Delta f_{90} = f_{90} - f_0$ was found to contain most of the information about the anisotropy, so we restrict our attention to this image alone. The Δf_{90} images for each band of EUVI-A and B are shown in Rows 1 and 3 of Fig. 4.17. The base differences $\Delta u_{90} = u_{90} - u_0$ for the corresponding deconvolved images are shown in Rows 2 and 4.

The Sun changes considerably during the 40 minutes between acquisition of f_0 and f_{90}, and most of the on-disk values of Δf_{90} are due to temporal variation. Regions far off the limb, however, are much less variable, and there the effects of stray light manifest as antipodal pairs of positive and negative regions forming an X-shaped pattern. In 171 and 195 Å the negative (blue) regions are found on the diagonal line 45° above the horizontal, and positive regions (red) are found on the line perpendicular. In 284 and 304 Å the pattern is reversed. The negative regions represent areas where the base image f_0 has more stray light than f_{90}, while the positive regions have less.

A simple interpretation of these observations is that the PSF wings have higher values along the axis defining the blue region, and lower values along the axis defining the red one. This observation prompted us to give the PSF elliptical cross sections with primary axes along the observed diagonals, as described in §4.2.2.2. The effectiveness of this model can be seen in the deconvolved differences Δu_{90}, where the X-shaped pattern is generally greatly reduced, although some artifacts remain. Note that the calibration roll data was not used to fit the PSF, and was acquired over four years after the early mission and lunar transit data that was used.

4.7.5 Pupil diffraction

Each EUVI instrument has four quarter annulus-shaped pupils, one for each filter band telescope, as shown in Fig. 4.18 (J.-P. Wuelser, *private communication*). It is shown below that Fraunhofer diffraction through each quarter annulus pupil has a PSF h^p with long-range anisotropic wings. Thus h^p could contribute significantly to the overall stray light distribution and possibly explain the anisotropy observed in calibration rolls. To test this hypothesis we compute h^p numerically and deconvolve it from the calibration rolls, finding that h^p by itself cannot account for the stray light anisotropy.

To describe the quarter annulus pupil shape mathematically, let $R_1 = 32.5\,\mathrm{mm}$ and $R_2 = 49.0\,\mathrm{mm}$ be the inner and outer radii of the annulus and $2b = 23.7\,\mathrm{mm}$ the horizontal distance between both the left and right annuli and the top and bottom annuli. Ignoring the mesh over the pupil, the pupil's transmittance function is

$$P(\xi) = \begin{cases} 1 & \text{if } R_1 \leq |\xi| \leq R_2 \text{ and } \xi_1, \xi_2 \geq b \\ 0 & \text{otherwise.} \end{cases} \tag{4.49}$$

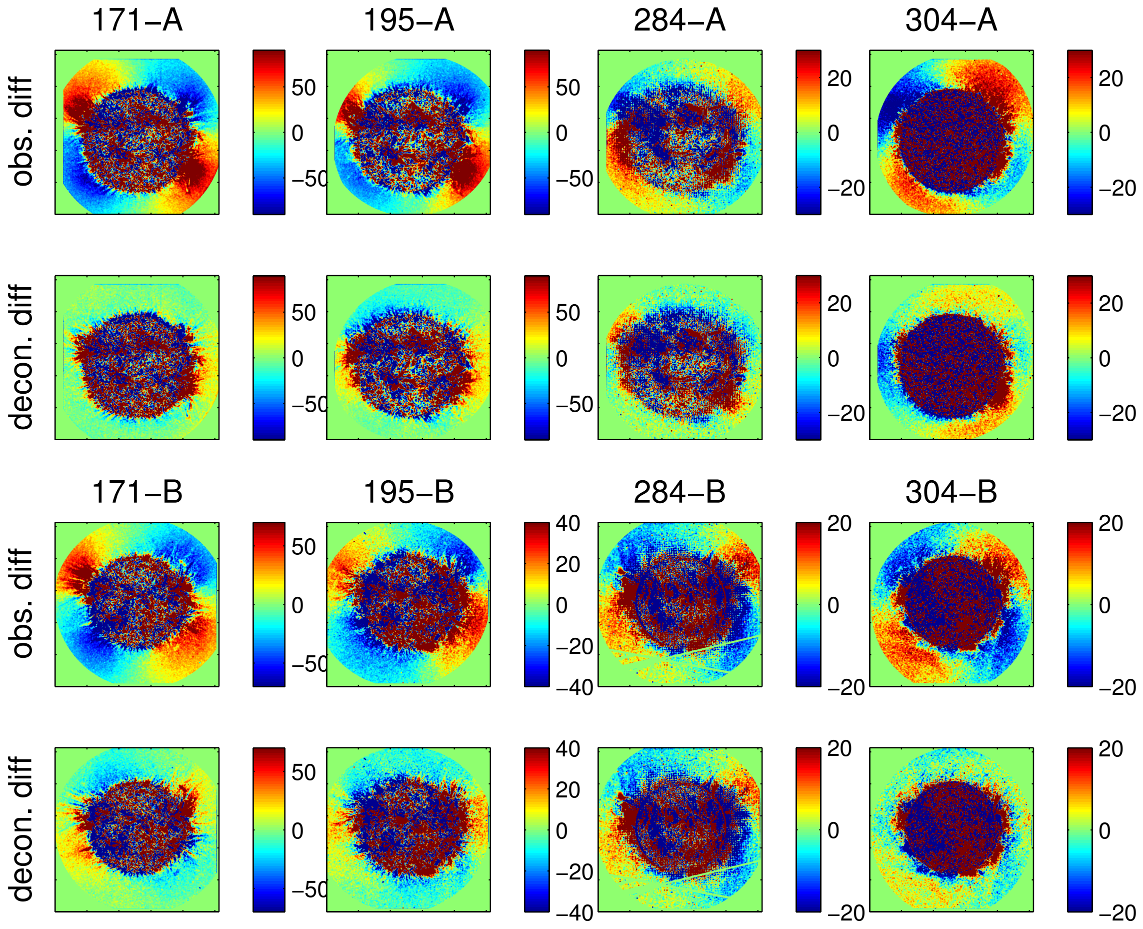

Figure 4.17: Analysis of calibration roll difference images in EUVI-A (top two rows) and B (bottom two rows) before and after stray light correction. All images are in units of DN, 4 × 4 binned, and an 8 × 8 moving average has been applied. Each roll image f_{90} has been rotated and shifted so the Sun's position matches the pre-roll image f_0. *Rows 1 and 3:* Difference $f_{90} - f_0$ of the pre-roll and 90° rolled images. *Rows 2 and 4:* Difference $u_{90} - u_0$ of deconvolved images.

The continuous PSF associated to Fraunhofer diffraction by this pupil is given by $H^p(x) \propto |\hat{P}(\tau x)|^2$, where $\tau = \Delta\theta_p/\lambda$. The corresponding discrete PSF $h^p(x)$ is obtained by integrating $H^p(x)$ over the area of a CCD pixel: for each pixel x in the image array,

$$h^p(x) = C \int_{[-1/2,1/2]^2} H^p(x + x')dx', \tag{4.50}$$

where C is a normalization constant. Given samples $H^p(j/Q_s)$ for $j \in \mathbb{Z}^2$, where Q_s is a positive odd number, we approximate $h^p(x)$ by discretizing the integral:

$$h^p(x) \approx C \sum_{-\lfloor Q_s/2 \rfloor \leq j_1, j_2 \leq \lfloor Q_s/2 \rfloor} H^p(x + j/Q_s) \cdot 1/Q_s^2, \tag{4.51}$$

where $\lfloor Q_s/2 \rfloor$ represents the largest integer less than $Q_s/2$. As we will see, $H^p(x)$ is highly oscillatory, and a large value of Q_s must be chosen to resolve the oscillations clearly and obtain an accurate integral approximation. We set $Q_s = 199$ in our computations.

We obtain the desired samples of H^p using the DFT. We define a square $[0, \overline{R}]^2$ in the pupil plane with $\overline{R} \geq R_2$ containing the full support of $P(\xi)$ and a sample spacing $\Delta\xi = \overline{R}/N_s$, and approximate the integral

$$\hat{P}(\tau x) = \int P(\xi) \exp(-2\pi i \tau \langle x, \xi \rangle) d\xi \tag{4.52}$$

by the discrete sum

$$\eta(x) = \sum_{k \in A} P(k\Delta\xi) \exp(-2\pi i \langle \tau x, k\Delta\xi \rangle) \Delta\xi^2 \tag{4.53}$$

where $A = \{0, \ldots, N_s - 1\}^2$. By sampling η at a spacing of $1/Q_s$ we obtain a DFT sum which can be calculated using the FFT algorithm:

$$\eta(j/Q_s) = \sum_{k \in A} P(k\Delta\xi) \exp(-2\pi i (\tau\Delta\xi/Q_s) \langle j, k \rangle) \Delta\xi^2 \quad \text{for } j \in \mathbb{Z}_{N_s}^2. \tag{4.54}$$

The definition of the DFT requires that $\tau\Delta\xi/Q_s = 1/N_s$, which simplifies to $\overline{R} = Q_s/\tau$. Setting $N_s = N_p Q_s$, where $N_p = 2048$ is the number of EUVI pixels, we substitute $|\eta|^2$ in for H^p in (4.51) to obtain

$$h^p(x) \approx C \sum_{-\lfloor Q_s/2 \rfloor \leq j_1, j_2 \leq \lfloor Q_s/2 \rfloor} |\eta(x + j/Q_s)|^2 \cdot 1/Q_s^2 \quad \text{for } x \in \mathbb{Z}_{N_p}^2, \tag{4.55}$$

where C is set by the sum-to-unity constraint, $\sum_{x \in \mathbb{Z}^2_{N_p}} h^p(x) = 1$. Put simply, the discrete pupil PSF h^p is approximated by binning down the array of samples of $|\eta|^2$ by a factor of Q_s in each dimension, then normalizing to unity.

The structure of the pupil diffraction pattern is clearest in the array $|\eta|^2$, before it is binned to form h^p. Views of the central portion of $|\eta|^2$ are shown in Fig. 4.18, *middle and right* for EUVI-A and B, 171 Å. It is symmetric with respect to 180° rotation, has two arms along the coordinate axes, and a fan-shaped component with a span from 15° to 75° above the horizontal axis in the first quadrant. The PSF h^p obtained after binning $|\eta|^2$ has the same features, but in a much lower resolution, and is not shown. The PSFs for the other bands are similar, except that those for 195 and 284 Å are obtained by reflection through the horizontal axis.

There are two sources of error in our calculation of h^p: error in discretizing the pixel integral and error due to aliasing in DFT approximation of the Fourier transform in (4.53). The high sampling rate of $|\eta|^2$ (nearly 40,000 samples per EUVI pixel) ensures the diffraction pattern is clearly resolved, and error in approximating the pixel integral to compute h^p is negligible. The contribution of aliasing can be bounded conservatively by calculating the PSF mass found outside a 1024×1024 square around the origin. Most aliasing will occur in this region, so if most of the mass is within the central region then aliasing must be minimal. In 171 Å this mass is $3.0 \cdot 10^{-5}$, a very small amount, and the other bands are similar.

To determine whether the fan-shaped component of h^p can explain the observed anisotropy in EUVI images, we deconvolve it from the calibration roll images for EUVI-B 171 Å. To ensure that the small amount of aliased energy in h^p cannot confound the results, we set to zero all pixels that are (1) outside the angles of 15° and 75° bounding the fan and (2) more than 5 pixels from the origin. This results in the purely fan-shaped PSF shown in Fig. 4.19, *right.* We then take the roll images f_0 and f_{90} and deconvolve them with h^p to obtain u^p_0 and u^p_{90}. Finally, the differences $f_{90} - f_0$ and $u^p_{90} - u^p_0$ are computed as in the previous section. These differences are shown in Fig. 4.19, *middle and right* respectively. It is clear that deconvolution with h^p has had no effect at all on the X-shaped pattern of anisotropy. However, the two antipodal fan shapes in h^p correspond quite well with the blue regions in $f_{90} - f_0$, which represent regions of heightened scatter. This correspondence is seen in all of the other bands as well (Fig. 4.17).

This analysis seems to rule out Fraunhofer diffraction by an ideal pupil as the cause of EUVI PSF anisotropy. However, there is a close correlation between the fan-shaped lobes of h^p and the observed anisotropy in calibration rolls. It is possible

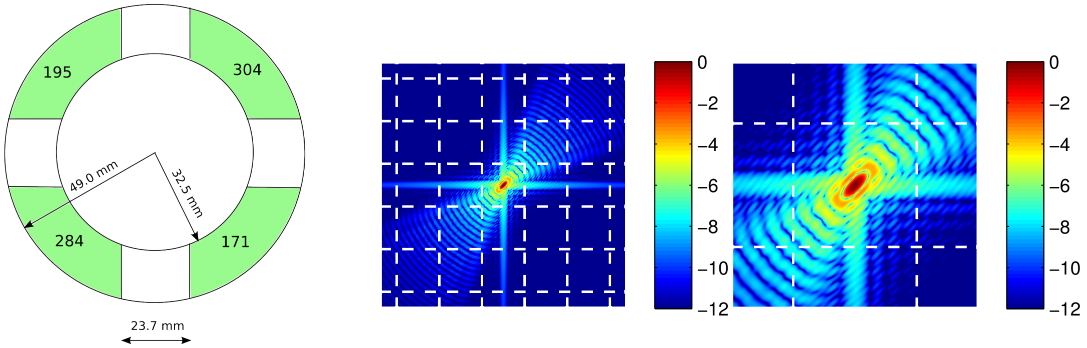

Figure 4.18: *Left:* The aperture mask over the EUVI primary mirror as seen from the Sun (J.-P. Wuelser, *private communication*). This mask defines the EUVI pupil. Ecliptic North is up on STEREO-A, and down on STEREO-B. *Middle:* Central portion of the array $|\eta|^2$ for EUVI-B 171 Å, logarithmic color scale. The values are normalized relative to the maximum at the core. White dotted lines are overlaid to show the scale of the EUVI pixel. *Right:* Enlarged view of the core of $|\eta|^2$.

that departures from the ideal quarter annulus shape, such as rough edges, could enhance the amount of scatter due to the pupil. Frensel propagation between the pupil, primary, and secondary mirrors, which we have neglected, may also enhance the scatter level.

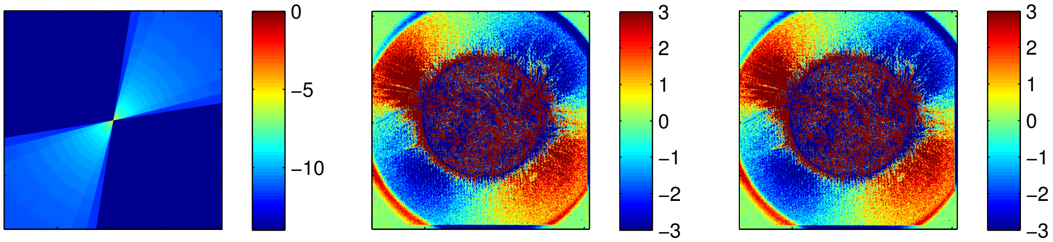

Figure 4.19: An experiment to determine whether pupil diffraction can account for the anisotropy observed in the EUVI-B 171 Å calibration roll data. The results are similar in other bands. *Left:* The full 2048 × 2048 pupil PSF h^p after all pixel values outside of the fan shape are set to zero (logarithmic color scale). *Middle:* The difference $f_{90} - f_0$ of coaligned calibration roll images. *Right:* The difference $u^p_{90} - u^p_0$ of images that have been deconvolved with the fan-shaped PSF at left.

CHAPTER 5

Conclusion

This thesis presented three self-contained contributions to the theory and applications of separable inverse problems, with a focus on the correction of image blur via blind or semiblind deconvolution. Here we summarize the most essential ideas of the thesis, potential impacts, and future directions.

5.1 Variable elimination, algorithms, and linear algebra

Chapter 2 describes how variable elimination may be generalized to solve optimization problems beyond least squares. Recall that variable elimination replaces the problem of minimizing a full cost function $F(y, z)$ with the problem of minimizing the reduced cost $F_r(y) = F(y, z_m(y))$, where $z_m(y)$ is the value of z that minimizes $F(y, z)$ given a fixed y value. Conventionally, variable elimination is viewed as a preliminary *algebraic* manipulation, reformulating the problem before any particular iterative method is applied to solve it. This chapter's key idea is that variable elimination can be accomplished through *algorithmic* manipulation, without ever requiring an expression for $z_m(y)$. This viewpoint enabled us to generalize variable elimination by formulating a semi-reduced method that accommodated bound constraints on z. More importantly, it enabled us to describe the precise algorithmic differences between full, semi-reduced, and reduced update methods, and predict when each would be most useful.

Surprisingly, most of the practical benefits came from using block Gaussian elimination to compute steps, while the utility of trial point adjustment seemed limited in our experiments. We have found that custom linear algebra routines can deliver substantial performance benefits in optimization, but few research or commercial codes provide a protocol for integrating novel linear algebra routines into their operation. The only code we have found to provide this functionality is [2], an interior point method for cone programming. But separable inverse problems are not cone programs and must be solved general nonlinear programming methods.

An object-oriented approach to implementing custom linear algebra routines in MATLAB optimization codes would be to define a linear operator class and overload the backslash operator to work with such linear operators. The method used to invert the operator would be specified by the user as part of a given operator. The SPOT toolbox [40] defines such a linear operator class and would provide a natural base for this development.

5.2 Camera shake correction by blind deconvolution

Chapter 3 describes a novel method for correcting camera shake by incremental sparse approximation. This method competes with the state of the art in deblurring performance on a standard test set, and learns blur kernels up to several times faster than state of the art methods. The main appeal of our method is its unity and simplicity of principle. Existing state-of-the-art methods tend to fall into two camps: joint MAP methods augmented with ad-hoc edge-finding heuristics, and kernel MAP methods which use an (approximate) variational Bayes technique to deal with a very difficult high-dimensional statistical inference problem. In contrast, our approach involves little more than an alternating projected gradient optimization with a gradually relaxed edge sparsity constraint.

The essential assumption of our method is that the most useful information about the blur kernel is contained near the strongest edges in the blurry image. This assumption works well for many images, but not all. Consider, for example, an image of a single strong edge blurred by a bimodal kernel. The bimodal kernel will cause the blurry image to have two edges of equal strength, our method will have difficulty determining which one is in the sharp image. This type of problem does sometimes occur in real images and blur kernels, and it can cause the method to fail. Addressing this issue seems to be an important direction for future work.

5.3 Stray light correction for extreme ultraviolet solar images

Chapter 4 proposes a solution to the stray light problem for the extreme ultraviolet imagers (EUVI) aboard the STEREO Ahead (A) and Behind (B) spacecraft, which are denoted EUVI-A and B. Extreme ultraviolet (EUV) images provide information about the coronal plasma and can be used to infer its density, temperature, and other characteristics via differential emission measure (DEM) analysis. Some of the most interesting structures, such as coronal holes and filament cavities, are much fainter than their surroundings and are heavily contaminated with stray light. These structures are involved in the generation of the solar wind and coronal mass ejections, and stray light correction is needed to study their governing physical processes. The PSFs we determined enable correction of stray light by a simple deconvolution procedure, and will become part of the SolarSoft preparation tools for EUVI images.

Variable elimination was the key mathematical tool enabling us to perform semi-blind deconvolution of the EUVI-B lunar transit images. We tried many methods to solve this problem, and the variable elimination method was by far the fastest and most robust. Some intuition for why variable elimination works so well is provided in the last two numerical examples of Chapter 2, where we study synthetic and toy variants of the solar stray light correction problem.

A major outstanding issue is the effect of compression. The wavelet-based ICER algorithm is used to compress EUVI images. In the brighter areas of an image, ICER has a benign and even useful denoising effect. In fainter regions, however, ICER can compromise accuracy because the low-pass wavelet coefficients containing large-scale intensity information may be quantized. In these cases, stray light correction can result in large negative regions off the limb. In the future we hope to quantify the uncertainties introduced by ICER and add them into the error analysis.

Substantial improvement of our PSFs may also be possible. In each band, there are artifacts in the deconvolved lunar transit images that are likely due to PSF inaccuracy. We expect that a more comprehensive physical modeling of the EUVI telescopes could yield a substantially improved PSF. An understanding of the physical origin of the PSF anisotropy would be a key milestone for such an effort.

Alternatively, one could posit a nonparametric model for the PSF in the hopes of identifying features that cannot be represented by our proposed parametric model. In fact, many of our preliminary efforts involved nonparametric modeling, and it was through these efforts that we first discovered the PSF anisotropy. Nonparametric modeling gives rise to much more challenging optimization problems and can be degraded by non-PSF effects such as bias variation. It is also possible that the PSF is spatially variant, even at large scales. If this is the case, a spatially variant nonparametric blind deconvolution would be required, and given the very large support of the PSF, this problem would be extremely difficult. We believe that future efforts to improve the PSF should rely on optical modeling as much as possible, invoking nonparametric elements only where there is very little basis to assume a parametric PSF model.

BIBLIOGRAPHY

[1] P. A. Absil and K. A. Gallivan. Accelerated line-search and trust-region methods. *SIAM Journal on Numerical Analysis*, 47(2):997–1018, 2009.

[2] M. Andersen, J. Dahl, Z. Liu, and L. Vandenberghe. Interior-point methods for large-scale cone programming. In *Optimization for Machine Learning*, pages 55–83. MIT Press, 2011.

[3] M. J. Aschwanden. *Physics of the Solar Corona. An Introduction with Problems and Solutions*. Springer, 2nd edition, 2006.

[4] H. Attouch, J. Bolte, and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, pages 1–39, 2011.

[5] F. Auchére and G. E. Artzner. EIT observations of the 15 November 1999 Mercury transit. *Solar Physics*, 219:217–230, 2004.

[6] G.R. Ayers and J.C. Dainty. Iterative blind deconvolution method and its applications. *Optics Letters*, 13(7):547–549, 1988.

[7] S. D. Babacan, R. Molina, M. N. Do, and A. K. Katsaggelos. Bayesian blind deconvolution with general sparse image priors. In *European Conference on Computer Vision (ECCV)*, Firenze, Italy, October 2012. Springer.

[8] J. M. Bardsley and C. R. Vogel. A nonnegatively constrained convex programming method for image reconstruction. *SIAM J. Sci. Comput.*, 25(4):1326–1343, 2004.

[9] M. Bertero, P. Boccacci, G. Desidera, and G. Vicidomini. Image deblurring with poisson data: from cells to galaxies. *Inverse Problems*, 25(12):123006, 2009.

[10] D.P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, 20(2):221–246, 1982.

[11] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, Massachusetts, 1999.

[12] E.G. Birgin, J.M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000.

[13] C.M. Bishop et al. *Pattern Recognition and Machine Learning.* Springer-Verlag New York, 2006.

[14] T. Blumensath and M.E. Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):298–309, 2010.

[15] Silvia Bonettini. Inexact block coordinate descent methods with application to non-negative matrix factorization. *IMA journal of numerical analysis*, 31(4):1431–1452, 2011.

[16] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[17] J. C. Brown, B. N. Dwivedi, et al. The interpretation of density sensitive line diagnostics from inhomogeneous plasmas. II - Non-isothermal plasmas. *Astronomy and Astrophysics*, 249(1):277, 1991.

[18] A. M. Buchanan and A. W. Fitzgibbon. Damped Newton algorithms for matrix factorization with missing data. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 316–322, 2005.

[19] D. S. Bunch, D. M. Gay, and R. E. Welsch. Algorithm 717: Subroutines for maximum likelihood and quasi-likelihood estimation of parameters in nonlinear regression models. *ACM Trans. Math. Softw.*, 19(1):109–130, March 1993.

[20] J.F. Cai, H. Ji, C. Liu, and Z. Shen. Framelet-based blind motion deblurring from a single image. *IEEE Transactions on Image Processing*, 21(2):562–572, 2012.

[21] P. Campisi and K. Egiazarian. *Blind Image Deconvolution: Theory and Applications.* CRC Press, 2007.

[22] T. F. Chan. An approximate Newton method for coupled nonlinear systems. *SIAM Journal on Numerical Analysis*, 22(5):904–913, 1985.

[23] T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7:370–375, March 1998.

[24] S. Cho and S. Lee. Fast motion deblurring. In *ACM Transactions on Graphics*, volume 28, page 145, 2009.

[25] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[26] J. Chung and J. G. Nagy. An efficient iterative approach for large-scale separable nonlinear inverse problems. *SIAM J. Sci. Comput.*, 31:4654–4674, January 2010.

[27] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, 2009.

[28] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[29] A. R. Conn, L. N. Vicente, and C. Visweswariah. Two-step algorithms for nonlinear optimization with structured applications. Research Report RC 21198(94689), IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY, 1998.

[30] Andrew R. Conn, Luis N. Vicente, and Chandu Visweswariah. Two-step algorithms for nonlinear optimization with structured applications. *SIAM Journal on Optimization*, 9(4):924–947, 1999.

[31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.

[32] A. Cornelio, E. Loli Piccolomini, and J. G. Nagy. Constrained variable projection method for blind deconvolution. *Journal of Physics: Conference Series*, 386(1):012005, 2012.

[33] T.A. Davis. Algorithm 915, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):8, 2011.

[34] C. E. DeForest, P. C. H. Martens, and M. J. Wills-Davey. Solar coronal structure and stray light in TRACE. *The Astrophysical Journal*, 690(2):1264, 2009.

[35] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, pages 272–279, 2008.

[36] Michael Elad, Mario A.T. Figueiredo, and Yi Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010.

[37] Y.-W. D. Fan and J. G. Nagy. An efficient computational approach for multi-frame blind deconvolution. *Journal of Computational and Applied Mathematics*, 236(8):2112–2125, 2012.

[38] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman. Removing camera shake from a single photograph. In *ACM Transactions on Graphics*, volume 25, pages 787–794, 2006.

[39] R. A. Frazin, A. M. Vásquez, and F. Kamalabadi. Quantitative, three-dimensional analysis of the global corona with multi-spacecraft differential emission measure tomography. *The Astrophysical Journal*, 701:547–560, 2009.

[40] M. P. Friedlander and E. van den Berg. Spot - a Linear-Operator Toolbox. `http://www.cs.ubc.ca/labs/scl/spot/`.

[41] E.M. Gafni and D.P. Bertsekas. Two-metric projection methods for constrained optimization. *SIAM Journal on Control and Optimization*, 22(6):936–964, 1984.

[42] R. Garg and R. Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 337–344. ACM, 2009.

[43] S. Gburek, J. Sylwester, and P. Martens. The TRACE telescope point spread function for the 171 Å filter. *Solar Physics*, 239:531–548, 2006.

[44] L. Gilles, C. R. Vogel, and J. M. Bardsley. Computational methods for a large-scale inverse problem arising in atmospheric optics. *Inverse Problems*, 18(1):237, 2002.

[45] Gene Golub and Victor Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19(2):R1, 2003.

[46] Gene H. Golub and Randall J. LeVeque. Extensions and uses of the variable projection algorithm for solving nonlinear least squares problems. In *Proceedings of the Army Numerical Analysis and Computing Conference, US Army Research Office, Washington DC*, pages 1–12, 1979.

[47] Gene H. Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on NumericalAnalysis*, 10(2):413–432, 1973.

[48] L. Golub and J.M. Pasachoff. *The Solar Corona.* Cambridge Univ. Press, 1997.

[49] J. W. Goodman. *Introduction to Fourier Optics.* McGraw-Hill, 2nd edition, 1996.

[50] J. W. Goodman. *Statistical Optics.* Wiley-Interscience, 1st edition, August 2000.

[51] L. Grippo and M. Sciandrone. Globally convergent block-coordinate techniques for unconstrained optimization. *Optimization Methods and Software*, 10(4):587–637, 1999.

[52] C. Guennou, F. Auchère, E. Soubrié, K. Bocchialini, and S. Parenti. On the accuracy of the differential emission measure diagnostics of solar plasmas. application to AIA/SDO. Part I: isothermal plasmas. *arXiv preprint arXiv:1210.2304*, 2012.

[53] P. C. Hansen, V. Pereyra, and G. Scherer. *Least Squares Data Fitting with Applications*. John Hopkins University Press, 2013.

[54] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2003.

[55] R. A. Howard et al. Sun Earth Connection Coronal and Heliospheric Investigation (SECCHI). *Space Sci. Rev.*, 136:67–115, 2008.

[56] Z. Huang, R. A. Frazin, E. Landi, W. B. Manchester, A. M. Vásquez, and T. I. Gombosi. Newly discovered global temperature structures in the quiet sun at solar minimum. *The Astrophysical Journal*, 755(2):86, 2012.

[57] M.W. Jacobson and J.A. Fessler. Joint estimation of image and deformation parameters in motion-corrected PET. In *Nuclear Science Symposium Conference Record, 2003 IEEE*, volume 5, pages 3290–3294 Vol.5, 2003.

[58] James R Janesick. *Scientific charge-coupled devices*, volume 117. SPIE Press, Bellingham, WA, 2001.

[59] M. Jin, W. B. Manchester, B. Van der Holst, J. R. Gruesbeck, R. A. Frazin, E. Landi, A. M. Vasquez, P. L. Lamy, A. Llebaria, A. Fedorov, et al. A global two-temperature corona and inner heliosphere model: A comprehensive validation study. *The Astrophysical Journal*, 745(1):6, 2012.

[60] L. Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT Numerical Mathematics*, 15:49–57, 1975.

[61] L. Kaufman and G. Sylvester. Separable nonlinear least squares with multiple right-hand sides. *SIAM Journal on Matrix Analysis and Applications*, 13(1):68–89, 1992.

[62] C.T. Kelley. *Iterative Methods for Optimization*. Frontiers in Applied Mathematics. SIAM, 1987.

[63] A. Kiely and M. Klimesh. The ICER progressive wavelet image compressor. *IPN Progress Report*, pages 42–155, 2003.

[64] Christof G. Krautschik, Masaaki Ito, Iwao Nishiyama, and Shinji Okazaki. Impact of EUV light scatter on CD control as a result of mask density changes. *Proc. SPIE*, 4688(1):289–301, 2002.

[65] A. S. Krieger, A. F. Timothy, and E. C. Roelof. A coronal hole and its identification as the source of a high velocity solar wind stream. *Solar Physics*, 29:505–525, April 1973.

[66] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 233–240, 2011.

[67] P.J. Lanzkron, D.J. Rose, and J.T. Wilkes. An analysis of approximate nonlinear elimination. *SIAM J. Sci. Comput.*, 17(2):538–559, 1996.

[68] D. R. Larson. The economy of photons. *Nature Methods*, 7(5):357–359, 05 2010.

[69] T. A. Laurence and B. A. Chromy. Efficient maximum likelihood estimator fitting of histograms. *Nature methods*, 7(5):338–339, 2010.

[70] A. Levin, R. Fergus, F. Durand, and W.T. Freeman. Deconvolution using natural image priors. *Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory*, 2007.

[71] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman. Understanding and evaluating blind deconvolution algorithms. *IEEE Conference on Computer Vision and Pattern Recognition*, 0:1964–1971, 2009.

[72] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2657–2664, 2011.

[73] M.E. Lopes. Estimating unknown sparsity in compressed sensing. *arXiv preprint arXiv:1204.4227*, 2012.

[74] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007.

[75] S. Mallat. *A Wavelet Tour of Signal Processing, 3rd ed., Third Edition: The Sparse Way.* Academic Press, December 2008.

[76] D. Martínez-Galarce, J. Harvey, et al. A novel forward-model technique for estimating EUV imaging performance: design and analysis of the SUVI telescope. *Proc. SPIE*, 7732(1):773237, 2010.

[77] M. Maus, M. Cotlet, et al. An experimental comparison of the maximum likelihood estimation and nonlinear least-squares fluorescence lifetime analysis of single molecules. *Analytical chemistry*, 73(9):2078–2086, 2001.

[78] K. M. Mullen and I. H. M. Van Stokkum. TIMP: an R package for modeling multi-way spectroscopic measurements. *Journal of Statistical Software*, 18(3):1–46, 2007.

[79] K. M. Mullen and I. H. M. van Stokkum. The variable projection algorithm in time-resolved spectroscopy, microscopy and mass spectrometry applications. *Numerical Algorithms*, 51:319–340, 2009.

[80] K. M. Mullen and I. H. M. van Stokkum. Sum-of-exponentials models for time-resolved spectroscopy data. In V. Pereyra and G. Scherer, editors, *Exponential Data Fitting and its Applications*, chapter 6. Bentham Science Publishers, Oak Park, 2010.

[81] D. Needell and J.A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[82] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag New York, 2nd edition, 2006.

[83] V.T. Olafsson, D.C. Noll, and J.A. Fessler. Fast joint reconstruction of dynamic R_2^* and field maps in functional MRI. *Medical Imaging, IEEE Transactions on*, 27(9):1177–1188, 2008.

[84] D.P. O'Leary and B.W. Rust. Variable projection for nonlinear least squares problems. *Computational Optimization and Applications*, pages 1–15, 2012.

[85] H. M. Ozaktas, M. A. Kutay, and Z. Zalevsky. *The fractional Fourier transform with applications in optics and signal processing*. Wiley New York, 2001.

[86] T.A. Parks. *Reducible nonlinear programming problems (separable least squares)*. PhD thesis, Rice University, 1985.

[87] V. Pereyra and G. Scherer, editors. *Exponential Data Fitting and its Applications*. Bentham Science Publishers, Oak Park, 2010.

[88] B. Poduval, C. E. DeForest, J. T. Schmelz, and S. Pathak. Point-spread Functions for the Extreme-ultraviolet Channels of SDO/AIA Telescopes. *The Astrophysical Journal*, 765:144, March 2013.

[89] F. S. G. Richards. A method of maximum-likelihood estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 23(2):pp. 469–475, 1961.

[90] A. Ruhe and P. Å. Wedin. Algorithms for separable nonlinear least squares problems. *SIAM Review*, 22(3):318–337, 1980.

[91] Y. Saad. *Iterative methods for sparse linear systems*, volume 620. PWS Publishing Company Boston, 1996.

[92] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for ℓ_1 regularization: A comparative study and two new approaches. *Machine Learning: ECML 2007*, pages 286–297, 2007.

[93] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM Transactions on Graphics*, volume 27, page 73, 2008.

[94] P. Shearer, R. A. Frazin, A. O. Hero III, and A. C. Gilbert. The first stray light corrected extreme-ultraviolet images of solar coronal holes. *The Astrophysical Journal Letters*, 749(1):L8, 2012.

[95] Paul Shearer and Anna C. Gilbert. A generalization of variable elimination for separable inverse problems beyond least squares. *Inverse Problems*, 29(4):045003, 2013.

[96] D. M. Sima and S. Van Huffel. Separable nonlinear least squares fitting with linear bound constraints and its application in magnetic resonance spectroscopy data quantification. *Journal of Computational and Applied Mathematics*, 203(1):264 – 278, 2007.

[97] G.K. Smyth. Partitioned algorithms for maximum likelihood and other nonlinear estimation. *Statistics and Computing*, 6(3):201–216, 1996.

[98] S. Sra, S. Nowozin, and S. J. Wright. Projected Newton-type methods in machine learning. In *Optimization for Machine Learning*, Neural Information Processing. MIT Press, 2011.

[99] J. L. Starck, E. Pantin, and F. Murtagh. Deconvolution in astronomy: A review. *PASP*, 114(800):1051–1069, 2002.

[100] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998.

[101] L.N. Trefethen and D. Bau III. *Numerical linear algebra*. SIAM, 1997.

[102] J.A. Tropp and A.C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[103] John R. Valenzuela, Jeffrey A. Fessler, and Richard G. Paxman. Joint estimation of stokes images and aberrations from phase-diverse polarimetric measurements. *J. Opt. Soc. Am. A*, 27(5):1185–1193, May 2010.

[104] B. van der Holst, W. B. Manchester IV, R. A. Frazin, A. M. Vásquez, G. Tóth, and T. I. Gombosi. A data-driven, two-temperature solar wind model with alfvén waves. *The Astrophysical Journal*, 725(1):1373, 2010.

[105] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.

[106] C. R. Vogel, T. F. Chan, and R. J. Plemmons. Fast algorithms for phase-diversity-based blind deconvolution. In D. Bonaccini and R. K. Tyson, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3353, pages 994–1005, September 1998.

[107] C. Wang, L.F. Sun, Z.Y. Chen, J.W. Zhang, and S.Q. Yang. Multi-scale blind motion deblurring using local minimum. *Inverse Problems*, 26(1):015003, 2009.

[108] R. W. M. Wedderburn. Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika*, 61(3):439–447, 1974.

[109] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 491–498, 2010.

[110] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. *European Conference on Computer Vision*, pages 157–170, 2010.

[111] K. Yamaoka, T. Nakagawa, and T. Uno. Application of Akaike's information criterion in the evaluation of linear pharmacokinetic equations. *Journal of Pharmacokinetics and Pharmacodynamics*, 6:165–175, 1978.

[112] Y. L. You and M. Kaveh. A regularization approach to joint blur identification and image restoration. *IEEE Transactions on Image Processing*, 5(3):416–28, 1996.

[113] F. Zhang. *The Schur Complement and its Applications*, volume 4 of *Numerical Methods and Algorithms*. Springer, 2005.

[114] H. Zhang and W.W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal on Optimization*, 14(4):1043–1056, 2004.