University of California

Los Angeles

Primal Dual Algorithms for Convex Models and Applications to Image Restoration, Registration and Nonlocal Inpainting

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mathematics

by

John Ernest Esser

2010

The dissertation of John Ernest Esser is approved.

Alan L. Yuille

Andrea L. Bertozzi

Luminita A. Vese

Tony F. Chan, Committee Chair

University of California, Los Angeles

2010

Table of Contents

List of Figures

List of Tables

Acknowledgments

First, I'd like to thank my advisor, Tony Chan, for all his support and advice. I especially appreciate the effort he's made to continue mentoring me and his other students after leaving UCLA for the NSF and even after becoming president of HKUST. His enthusiasm and unlimited energy are inspiring.

Many thanks to Xiaoqun Zhang for the numerous valuable discussions about the algorithms and applications presented here. Most of the material in Chapter 3 is based on [EZC09], written in collaboration with Xiaoqun and Tony. I especially thank Xiaoqun for contributing the convergence results for the special cases of PDHG discussed in Section 3.5.2.1 and for sharing her convergence analysis of the split inexact Uzawa method [ZBO09], which plays an important role in many of the examples presented here.

I'd like to thank Berta Sandberg for lots of helpful discussions, advice and even some C++ code that helped me get started when I was first beginning to do image processing related research.

Thanks to Andrea Bertozzi for her support and for including me in her research group after Tony left for the NSF. Although I've changed research directions since then, I learned a lot from her advice on various projects.

Thanks to Luminita Vese for advice and helpful discussions about several projects including the convex registration model which appears in Chapter 4. I also learned a lot from her seminars and graduate courses on image processing and optimization.

I'd also like to thank Stan Osher for including me in his research group, pointing out relevant recent papers and for the optimization class in Winter 2009,

which helped clarify some of the connections between algorithms I was writing about in [Ess09] and that are discussed in Chapter 2.

Thanks to Paul Tseng for pointing out several key references and providing the simple proof of the general Moreau decomposition (Throrem 2.3.1).

I'd like to additionally thank Jeremy Brandman, Ethan Brown, Jerome Darbon, Xavier Bresson, Mingqiang Zhu and Tom Goldstein for helpful discussions that improved the quality of this work. Thanks also to Jeremy for proofreading and helping improve the exposition of [Ess09].

I'm extremely grateful to my parents, Doug and Rita Esser, for their constant support and encouragement. I'd like to thank my brother, Mike, for lending an ear even at odd hours of the night, spamming memes and offering his unique perspective on things. I also thank my cousin, Teresa, for the unlimited supply of coffee which has kept me humming for years and fueled much of this work.

The research presented in this dissertation was supported at various times by the following grants: ONR N00014-03-1-0071, NSF DMS-0610079, NSF CCF-0528583 and NSF DMS-0312222.

Vita

1980 Born, Seattle, Washington

1998-2003 University of Washington:

- B.S. Mathematics with College Honors
- B.S. Applied and Computational Mathematical Sciences
- B.A. Italian
- Minor: Physics

2003-2010 UCLA:

- M.A. Mathematics, 2004
- Ph.D. Mathematics, February 2010

Publications

Ernie Esser, *Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman*, April 2009
ftp://ftp.math.ucla.edu/pub/camreport/cam09-31.pdf

Ernie Esser, Xiaoqun Zhang and Tony Chan, *A General Framework for a Class of First Order Primal-Dual Algorithms for TV Minimization*, August 2009

ftp://ftp.math.ucla.edu/pub/camreport/cam09-67.pdf (submitted to SIIMS)

Ernie Esser, *A Convex Model for Image Registration*, January 2010
ftp://ftp.math.ucla.edu/pub/camreport/cam10-04.pdf

Abstract of the Dissertation

Primal Dual Algorithms for Convex Models and Applications to Image Restoration, Registration and Nonlocal Inpainting

by

John Ernest Esser

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2010

Professor Tony F. Chan, Chair

The main subject of this dissertation is a class of practical algorithms for minimizing convex non-differentiable functionals coming from image processing problems defined as variational models. This work builds largely on the work of Goldstein and Osher [GO09] and Zhu and Chan [ZC08] who proposed respectively the split Bregman and the primal dual hybrid gradient (PDHG) methods for total variation (TV) image restoration. We relate these algorithms to classical methods and generalize their applicability. We also propose new convex variational models for image registration and patch-based nonlocal inpainting and solve them with a variant of the PDHG method.

We draw connections between popular methods for convex optimization in image processing by putting them in a general framework of Lagrangian-based alternating direction methods. Furthermore, operator splitting and decomposition techniques are used to generalize their application to a large class of problems, namely minimizing sums of convex functions composed with linear operators and subject to convex constraints. Numerous problems in image and signal process-

ing such as denoising, deblurring, basis pursuit, segmentation, inpainting and many more can be modeled as minimizing exactly such functionals. Numerical examples focus especially on when it is possible to minimize such functionals by solving a sequence of simple convex minimization problems with explicit formulas for their solutions.

In the case of the split Bregman method, we point out an equivalence to the classical alternating direction method of multipliers (ADMM) and Douglas Rachford splitting methods. Existing convergence arguments and some minor extensions justify application to common image processing problems.

In the case of PDHG, its general convergence is still an open problem, but in joint work with Xiaoqun Zhang and Tony Chan we propose a simple modification that guarantees convergence. We also show convergence of some special cases of the original method. Numerical examples show PDHG and its variants to be especially well suited for large scale problems because their simple, explicit iterations can be constructed to avoid the need to invert large matrices at each iteration.

The two proposed convex variational models for image registration and nonlocal inpainting are novel because most existing variational approaches require minimizing nonconvex functionals.

CHAPTER 1

Introduction

An important class of problems in image processing, and now also compressive sensing, is convex programs involving l_1 or total variation (TV) minimization. The use of l_1 and TV regularizers has been shown to be very effective in regularizing inverse problems where one expects the recovered image or signal to be sparse or piecewise constant. The l_1 norm encourages sparsity of the signal while the TV seminorm encourages sparsity of the gradient. Illustrative examples include ROF denoising [ROF92] and basis pursuit [CDS98]. A lack of differentiability makes minimizing such functionals computationally challenging, and so there is considerable interest in efficient algorithms, especially for large scale problems. There is an additional need for algorithms that can efficiently take advantage of the separable structure of more complicated models consisting of sums of convex functionals composed with linear operators and subject to convex constraints. Algorithms such as split Bregman [GO09], the split Inexact Uzawa method [ZBO09] and the primal dual hybrid gradient (PDHG) method [ZC08] have been shown to yield simple, fast and effective algorithms for these types of problems. These recent algorithms also have many interesting connections to classical Lagrangian methods for the general problem of minimizing sums of convex functionals subject to linear equality constraints. There are close connections for example to the alternating direction method of multipliers (ADMM) [BT89, EB92, GM76, GM75] and the alternating minimization algorithm (AMA) of [Tse91]. These algorithms

can be especially effective when the convex functionals are based on l_2 and l_1-like norms. Connections between these algorithms as well as the operator splitting techniques that allow them to be effectively applied are discussed in Chapters 2 and 3.

Chapter 2 is based largely on the paper, *Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman* [Ess09]. We show that analogous to the connection between Bregman iteration and the method of multipliers [Hes69, Pop80] that was pointed out in [YOG08], a similar connection can be made between the split Bregman algorithm and ADMM. Existing convergence theory for ADMM [EB92] can therefore be used to justify both the alternating step and inexact minimizations used in split Bregman for the cases in which the algorithms are equivalent. Application of these algorithms to different image processing problems is simplified by rewriting these problems in a general form that still includes constrained and unconstrained TV and l_1 minimization as was investigated in [GO09]. Numerical results for the application to TV-l_1 minimization [CEN06] are presented. The dual interpretation of ADMM as Douglas Rachford splitting applied to a dual problem is well studied [Gab83, GT89, Eck89, EB92, LM79], and we examine this dual interpretation in some special cases. We also discuss applications of several related methods including AMA and the split inexact Uzawa method of [ZBO09], which are sometimes better suited for problems where further decoupling of variables is useful.

Chapter 3 is based on the paper, *A General Framework for a Class of First Order Primal-Dual Algorithms for TV Minimization* [EZC09], which represents joint work with Xiaoqun Zhang and Tony Chan. We generalize the primal-dual hybrid gradient (PDHG) algorithm proposed by Zhu and Chan in [ZC08], draw connections to similar methods and discuss convergence of several special cases

and modifications. In particular, we point out a convergence result for a modified version of PDHG that has a similarly good empirical convergence rate for total variation minimization problems. Its convergence follows from interpreting it as the split inexact Uzawa method discussed in [ZBO09]. We also prove a convergence result for PDHG applied to TV denoising with some restrictions on the PDHG step size parameters. It is shown how to interpret this special case as a projected averaged gradient method applied to the dual functional. We discuss the range of parameters for which the inexact Uzawa method and the projected averaged gradient method can be shown to converge. We also present some numerical results for these algorithms applied to TV denoising, TV deblurring, constrained l_1 minimization and multiphase segmentation problems. The effectiveness of the modified PDHG method for large scale, non-differentiable convex problems is further demonstrated in Chapters 4 and 5 where it is successfully applied to convex models for image registration and nonlocal inpainting.

Chapter 4 is based on the paper, *A Convex Model for Image Registration* [Ess10]. Variational methods for image registration generally involve minimizing a nonconvex functional with respect to the unknown displacement between two given images. A linear approximation of the image intensities is often used to obtain a convex approximation to the model, but it is only valid for small deformations. Algorithms such as gradient descent can get stuck in undesirable local minima of the nonconvex functional. Here, instead of seeking a global minimum of a nonconvex functional, and without making a small deformation assumption, we introduce and work with a different, convex model for the registration problem. In particular we consider a graph-based formulation that requires minimizing a convex function on the edges of the graph instead of working directly with the displacement field. The corresponding displacement can be inferred from the edge function. The convex model generally involves many more variables, but its

global minimum can be a better solution than a local minimum of the nonconvex model. We use a convergent variant of the PDHG algorithm for the numerical examples.

In Chapter 5 we propose a convex variational model for nonlocal image inpainting that uses nonlocal image patches to fill in a large missing region in a manner consistent with its boundary. Existing convex inpainting models, such as TV inpainting [CS05] tend to be based on propagating local information into the unknown region and therefore aren't always well suited for filling in areas far from the boundary. Usually greedy approaches are employed for exemplar-based inpainting similar to the texture synthesis technique of [EL99]. Previous variational methods for nonlocal texture inpainting have also been proposed [DSC03, ALM08, ACS09, ZC09], but they are all based on nonconvex models. Convexity in the proposed model is achieved by allowing unknown patches overlapping the inpainting region to be weighted averages of known image patches. It's possible to express the proposed functional solely in terms of these weights. The functional consists of a data fidelity term that encourages the unknown patches to agree with known boundary data and other patches that they overlap and a regularizing term to encourage spatial correspondence between the unknown patches and the known patches they are weighted averages of. A nonconvex modification to the functional is also proposed to promote greater sparsity of the weights when needed. Again we use a variant of PDHG to compute the numerical examples.

The main contributions of this dissertation are summarized in the following list:

- Described a general framework for a class of primal-dual algorithms that explains the connections between PDHG, ADMM, Douglas Rachford splitting, AMA, proximal forward backward splitting and split inexact Uzawa methods (Chapter 3, Figure 3.1)
- Clarified the convergence of split Bregman via its connection to ADMM (Section 2.3.2.3)
- Proposed a modification of the PDHG algorithm that converges by an equivalence to the split inexact Uzawa method (Section 3.4.2.4)
- Discussed operator splitting techniques for applying ADMM, PDHG and their variants to a large class of convex models (Sections 2.1 and 3.6.1)
- Used a generalized Moreau decomposition to explain the dual interpretations of ADMM, AMA and split inexact Uzawa (Sections 2.3.2.2, 3.4.2.1 and 3.4.2.4)
- Explained both primal and dual derivations for general shrinkage (soft thresholding) formulas (Section 2.4.2)
- Proposed a convex model for image registration (Chapter 4)
- Proposed a convex model for nonlocal patch-based image inpainting (Chapter 5)
- Demonstrated the successful application of ADMM and PDHG variants to image restoration, multiphase segmentation and the proposed registration and nonlocal inpainting models

Table 1.1: Summary of contributions

CHAPTER 2

Connection Between Split Bregman and ADMM and Applications to Convex Programs with Separable Structure

2.1 Introduction

There is extensive literature in convex optimization and numerical analysis about splitting methods for minimizing a sum of two convex functions subject to linear equality constraints. A general form of such a problem is

$$\begin{aligned} \min_{z \in \mathbb{R}^n, u \in \mathbb{R}^m} \quad & F(z) + H(u), \\ Bz + Au = b \end{aligned} \tag{P0}$$

where $F : \mathbb{R}^n \to (-\infty, \infty]$ and $H : \mathbb{R}^m \to (-\infty, \infty]$ are closed proper convex functions, A is a $d \times m$ matrix, B is a $d \times n$ matrix and $b \in \mathbb{R}^d$. Many variational models in image processing consist of minimizing sums of convex functionals composed with linear operators and subject to convex constraints. Such problems often have the form

$$\begin{aligned} \min_{u \in \mathbb{R}^m} \quad & J(u), \\ Ku = f \end{aligned} \tag{2.1}$$

where $J(u)$ has separable structure in the sense that it can be written as a sum of closed proper convex functions H and G_i,

$$J(u) = H(u) + \sum_{i=1}^{N} G_i(A_i u + b_i).$$

Additional convex constraints besides linear equality constraints can be incorporated into the functional by way of convex indicator functions. For example, to constrain u to a convex set S, one could define H or one of the G_i terms to equal the convex indicator function g_S for S defined by

$$g_S(u) = \begin{cases} 0 & \text{if } u \in S \\ \infty & \text{otherwise.} \end{cases}$$

The separable structure of the convex program in (2.1) allows it to be written in the form of (P0). To see how, suppose $G_i : \mathbb{R}^{n_i} \to (-\infty, \infty]$, $f \in \mathbb{R}^s$, $b_i \in \mathbb{R}^{n_i}$, each A_i is a $n_i \times m$ matrix and K is a $s \times m$ matrix. An equivalent formulation that decouples the G_i is obtained by introducing new variables z_i and constraints $z_i = A_i u + b_i$. This can be written in the form of (P0) by letting $F(z) = \sum_{i=1}^{N} G_i(z_i)$, $n = \sum_{i=1}^{N} n_i$, $z = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}$, $B = \begin{bmatrix} -I \\ 0 \end{bmatrix}$, $A = \begin{bmatrix} A_1 \\ \vdots \\ A_N \\ K \end{bmatrix}$, and $b = \begin{bmatrix} -b_1 \\ \vdots \\ -b_N \\ f \end{bmatrix}$. Letting $d = n + s$, note that A is a $d \times m$ matrix, B is a $d \times n$ matrix and $b \in \mathbb{R}^d$.

By the above equivalence, classical splitting methods for solving P0 can be straightforwardly applied to problems of the form (2.1). Similar decomposition strategies are discussed for example in [BT89], [Ber99], [Roc70] and [Tse91]. The goal is to produce algorithms that consist of simple, easy to compute steps that can deal with the terms of $J(u)$ one at a time. One approach based on duality leads to augmented Lagrangian type methods that can be interpreted as splitting methods applied to a dual formulation of the problem. A good summary

of these methods can be found in chapter three of [GT89] and Eckstein's thesis [Eck89]. Here we will focus mainly on ADMM because of its connection to the Split Bregman algorithm of Goldstein and Osher. They show in [GO09] how to simplify the minimization of convex functionals of u involving the l_1 norm of a convex function $\Phi(u)$. They replace $\Phi(u)$ with a new variable z, add a constraint $z = \Phi(u)$ and then use Bregman iteration [YOG08] techniques to handle the resulting constrained optimization problem. A key application is functionals containing $\|u\|_{TV}$. A related splitting approach that uses continuation methods to handle the constraints has been studied by Wang, Yin and Zhang, [WYZ07] and applied to TV minimization problems including TV-l_1 ([GLN08], [YZY09]). The connection between Bregman iteration and the augmented Lagrangian for constrained optimization problems with linear equality constraints is discussed by Yin, Osher, Goldfarb and Darbon in [YOG08]. They show Bregman iteration is equivalent to the method of multipliers of Hestenes [Hes69] and Powell [Pop80] when the constraints are linear. The augmented Lagrangian for problem (2.1) is

$$L_\alpha(u, \lambda) = J(u) + \langle \lambda, f - Ku \rangle + \frac{\alpha}{2}\|f - Ku\|^2,$$

where $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ denote the Euclidean norm and standard inner product. The method of multipliers is to iterate

$$\begin{aligned} u^{k+1} &= \arg\min_{u \in \mathbb{R}^m} L_\alpha(u, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + \alpha(f - Ku^{k+1}), \end{aligned} \tag{2.2}$$

whereas Bregman iteration yields

$$\begin{aligned} u^{k+1} &= \arg\min_{u \in \mathbb{R}^m} J(u) - J(u^k) - \langle p^k, u - u^k \rangle + \frac{\alpha}{2}\|f - Ku\|^2 \\ p^{k+1} &= p^k + \alpha K^T(f - Ku^{k+1}). \end{aligned} \tag{2.3}$$

$J(u)-J(u^k)-\langle p^k, u-u^k\rangle$ is the Bregman distance between u and u^k, where p^k is a subgradient of J at u^k. Similarly, in the special case when Φ is linear, an interpretation of the split Bregman algorithm, explained in sections 2.3.1.1 and 2.3.2.1, is to alternately minimize with respect to u and z the augmented Lagrangian associated to the constrained problem and then to update a Lagrange multiplier. This procedure also describes ADMM, which goes back to Glowinski and Marocco [GM75], and Gabay and Mercier [GM76]. The augmented Lagrangian for problem (P0) is

$$L_\alpha(z,u,\lambda) = F(z) + H(u) + \langle \lambda, b - Au - Bz\rangle + \frac{\alpha}{2}\|b - Au - Bz\|^2,$$

and the ADMM iterations are given by

$$\begin{aligned} z^{k+1} &= \arg\min_{z\in\mathbb{R}^n} L_\alpha(z, u^k, \lambda^k) \\ u^{k+1} &= \arg\min_{u\in\mathbb{R}^m} L_\alpha(z^{k+1}, u, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}). \end{aligned} \tag{2.4}$$

ADMM can also be interpreted as Douglas Rachford splitting [DR56] applied to the dual problem. The connection between these two interpretations was first explored by Gabay [Gab83] and is also discussed by Glowinski and Le Tallec in [GT89]. The dual version of the algorithm was studied by Lions and Mercier [LM79]. The equivalence of ADMM to a proximal point method was studied by Eckstein and Bertsekas [EB92], who also generalized the convergence theory to allow for inexact minimizations. Direct convergence proofs in the exact minimization case can also be found for example in [GT89, BT89, WT09]. Techniques regarding applying ADMM to problems with separable structure can be found for example in [FG83] and are discussed in detail by Bertsekas and Tsitsiklis in ([BT89] Section 3.4.4). The connection between split Bregman and Douglas Rachford splitting has also been made by Setzer [Set09].

Other splitting methods besides Douglas Rachford splitting can be applied to the dual problem, which is a special case of the well studied more general problem of finding a zero of the sum of two maximal monotone operators. See for example [Eck89] and [LM79]. Some splitting methods applied to the dual problem can also be interpreted in terms of alternating minimization of the augmented Lagrangian. For example, Peaceman Rachford splitting [PR55] corresponds to an alternating minimization algorithm very similar to ADMM except that it updates the Lagrange multiplier twice, once after each minimization of the augmented Lagrangian [GT89].

Proximal forward backward splitting can also be effectively applied to the dual problem. This splitting procedure, which goes back to Lions and Mercier [LM79] and Passty [Pas79], appears in many applications. Some examples include classical methods such as gradient projection and more recent ones such as the iterative thresholding algorithm FPC of Hale, Yin and Zhang [HYZ07] and the framelet inpainting algorithm of Cai, Chan and Shen [CCS08].

The Lagrangian interpretation of the dual application of forward backward splitting was studied by Tseng in [Tse91]. He shows that it corresponds to an algorithm with the same steps as ADMM except that one of the minimizations of the augmented Lagrangian, $L_\alpha(z, u, \lambda)$, is replaced by minimization of the Lagrangian, which for (P0) is

$$L(z, u, \lambda) = F(z) + H(u) + \langle \lambda, b - Au - Bz \rangle.$$

The resulting iterations are given by

$$\begin{aligned} u^{k+1} &= \arg\min_{u \in \mathbb{R}^m} L(z^k, u, \lambda^k) \\ z^{k+1} &= \arg\min_{z \in \mathbb{R}^n} L_\alpha(z, u^{k+1}, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}). \end{aligned} \tag{2.5}$$

Tseng called this the alternating minimization algorithm, referred to in shorthand as AMA. This method is useful for solving (P0) when H is strictly convex but including the augmented quadratic penalty leads to a minimization step that is more difficult to solve.

There are other methods for decoupling variables that don't require the functional to be strictly convex. An example is the predictor corrector proximal method (PCPM) by Chen and Teboulle [CT94], which alternates proximal steps for the primal and dual variables. The PCPM iterations are given by

$$\begin{aligned}
u^{k+1} &= \arg\min_{u\in\mathbb{R}^m} L(z^k, u, \lambda^k) + \frac{1}{2\alpha_k}\|u-u^k\|^2 \\
z^{k+1} &= \arg\min_{z\in\mathbb{R}^n} L(z, u^k, \lambda^k) + \frac{1}{2\alpha_k}\|z-z^k\|^2 \\
\lambda^{k+1} &= \lambda^k + (\alpha_{k+1}+\alpha_k)(b-Au^{k+1}-Bz^{k+1}) - \alpha_k(b-Au^k-Bz^k).
\end{aligned}$$

This method can require many iterations. Another technique to undo the coupling of variables that results from quadratic penalty terms of the form $\frac{\alpha_k}{2}\|Ku-f\|^2$ is to replace such a penalty with one of the form $\frac{1}{2\delta_k}\|u-u^k+\alpha_k\delta_k K^T(Ku^k-f)\|^2$, which instead penalizes the distance of u from a linearization of the original penalty. This was applied to the method of multipliers by Stephanopoulos and Westerberg in [SW75]. It was used in the derivation of the linearized Bregman algorithm in [YOG08]. This technique is also used with Bregman iteration methods by Zhang, Burger, Bresson and Osher in [ZBB09], leading to the Bregman Operator Splitting (BOS) algorithm, which they apply for example to nonlocal TV minimization problems. They also show the connection to inexact Uzawa methods. Written as an inexact Uzawa method, the BOS algorithm applied to

(2.1) yields the iterations

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} J(u) + \langle \lambda^k, f - Ku\rangle + \frac{1}{2\delta_k}\|u - u^k + \alpha_k\delta_k K^T(Ku^k - f)\|^2$$
$$\lambda^{k+1} = \lambda^k + \alpha_k(f - Ku^{k+1}). \tag{2.6}$$

This decoupling idea was extended to to splitting applications by Zhang, Burger and Osher in ([ZBO09], Algorithm A_1). It can be applied to (P0) by adding an additional quadratic penalty to each minimization step of ADMM (2.4). The resulting method will be referred to here by the split inexact Uzawa method. The iterations when applied to (P0) are given by

$$\begin{aligned} z^{k+1} &= \arg\min_{z\in\mathbb{R}^n} L_\alpha(z, u^k, \lambda^k) + \frac{1}{2}\|z - z^k\|^2_{Q_1} \\ u^{k+1} &= \arg\min_{u\in\mathbb{R}^m} L_\alpha(z^{k+1}, u, \lambda^k) + \frac{1}{2}\|u - u^k\|^2_{Q_2} \\ \lambda^{k+1} &= \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}), \end{aligned} \tag{2.7}$$

where Q_1, Q_2 are positive definite matrices and $\|z\|^2_{Q_1} = \langle Q_1 z, z\rangle$, $\|u\|^2_{Q_2} = \langle Q_2 u, u\rangle$. Although Q_1 and Q_2 can be arbitrary positive definite matrices, they can also be chosen to effectively linearize the quadratic penalties in the ADMM minimization steps by letting $Q_1 = \frac{1}{\delta} - \alpha B^T B$ and $Q_2 = \frac{1}{\delta} - \alpha A^T A$, with δ and α chosen to ensure positive definiteness. An example of this application is given in Section 2.3.3.2 and it's connection to a variant of the PDHG method is discussed in Chapter 3.

This chapter consists of three main parts. The first part discusses the Lagrangian formulation of the problem (P0) and the dual problem. The second part focuses on exploring the connection between split Bregman and ADMM, their application to (P0) and their dual interpretation. It also demonstrates how further decoupling of variables is possible using AMA and BOS. The third part

shows how to apply these algorithms to some example image processing problems, focusing on applications that illustrate how to take advantage of problems' separable structure.

2.2 The Primal and Dual Problems

Lagrangian duality will play an important role in the analysis of (P0). In this section we define a Lagrangian formulation of (P0) and the dual problem. We also discuss conditions that guarantee solutions to the primal and dual problems.

2.2.1 Lagrangian Formulation and Dual Problem

Associated to the primal problem (P0) is the Lagrangian

$$L(z, u, \lambda) = F(z) + H(u) + \langle \lambda, b - Au - Bz \rangle, \tag{2.8}$$

where the dual variable $\lambda \in \mathbb{R}^d$ can be thought of as a vector of Lagrange multipliers. The dual functional $q(\lambda)$ is a concave function $q : \mathbb{R}^d \to [-\infty, \infty)$ defined by

$$q(\lambda) = \inf_{u \in \mathbb{R}^m, z \in \mathbb{R}^n} L(z, u, \lambda). \tag{2.9}$$

The dual problem to (P0) is

$$\max_{\lambda \in \mathbb{R}^d} q(\lambda). \tag{D0}$$

Since (P0) is a convex programming problem with linear constraints, if it has an optimal solution (z^*, u^*) then (D0) also has an optimal solution λ^*, and

$$F(z^*) + H(u^*) = q(\lambda^*),$$

which is to say that the duality gap is zero, ([Ber99] 5.2, [Roc70] 28.2, 28.4). To guarantee existence of an optimal solution to (P0), assume that the set

$$\{(z, u) : F(z) + H(u) \leq c \ , \ Au + Bz = b\}$$

is nonempty and bounded for some $c \in \mathbb{R}$. Alternatively, we could assume that $Ku = f$ has a solution, and if it's not unique, which it probably won't be, then assume $F(z) + H(u)$ is coercive on the affine subspace defined by $Au + Bz = b$. Either way, we can equivalently minimize over a compact subset. Since F and H are closed proper convex functions, which is to say lower semicontinuous convex functions not identically infinity, Weierstrass' theorem implies a minimum is attained [Ber99].

2.2.2 Saddle Point Formulation and Optimality Conditions

Finding optimal solutions of (P0) and (D0) is equivalent to finding a saddle point of L. More precisely ([Roc70] 28.3), (z^*, u^*) is an optimal primal solution and λ^* is an optimal dual solution if and only if

$$L(z^*, u^*, \lambda) \leq L(z^*, u^*, \lambda^*) \leq L(z, u, \lambda^*) \qquad \forall\ z, u, \lambda. \tag{2.10}$$

From this it follows that

$$\begin{aligned}
&\max_{\lambda \in \mathbb{R}^d} F(z^*) + H(u^*) + \langle \lambda, b - Au^* - Bz^* \rangle = L(z^*, u^*, \lambda^*) \\
&= \min_{u \in \mathbb{R}^m, z \in \mathbb{R}^n} F(z) + H(u) + \langle \lambda^*, b - Au - Bz \rangle,
\end{aligned}$$

from which we can directly read off the Kuhn-Tucker optimality conditions.

$$Au^* + Bz^* = b \tag{2.11a}$$

$$B^T \lambda^* \in \partial F(z^*) \tag{2.11b}$$

$$A^T \lambda^* \in \partial H(u^*), \tag{2.11c}$$

where ∂ denotes the subdifferential, defined by

$$\partial F(z^*) = \left\{ p \in \mathbb{R}^n : F(v) \geq F(z^*) + \langle p, v - z^* \rangle\ \forall v \in \mathbb{R}^n \right\},$$

$$\partial H(u^*) = \{q \in \mathbb{R}^m : H(w) \geq H(u^*) + \langle q, w - u^* \rangle \ \forall w \in \mathbb{R}^m\}.$$

These optimality conditions (2.11) hold if and only if (z^*, u^*, λ^*) is a saddle point for L ([Roc70] 28.3). Note also that $L(z^*, u^*, \lambda^*) = F(z^*) + H(u^*)$.

2.2.3 Dual Functional

The dual functional $q(\lambda)$ (2.9) can be written in terms of the Legendre-Fenchel transforms of F and H.

$$\begin{aligned}
q(\lambda) &= \inf_{z \in \mathbb{R}^n, u \in \mathbb{R}^m} F(z) + \langle \lambda, b - Bz - Au \rangle + H(u) \\
&= \inf_{z \in \mathbb{R}^n} \left(F(z) - \langle \lambda, Bz \rangle\right) + \inf_{u \in \mathbb{R}^m} \left(H(u) - \langle \lambda, Au \rangle\right) + \langle \lambda, b \rangle \\
&= -\sup_{z \in \mathbb{R}^n} \left(\langle B^T\lambda, z \rangle - F(z)\right) - \sup_{u \in \mathbb{R}^m} \left(\langle A^T\lambda, u \rangle - H(u)\right) + \langle \lambda, b \rangle \\
&= -F^*(B^T\lambda) - H^*(A^T\lambda) + \langle \lambda, b \rangle,
\end{aligned}$$

where F^* and H^* denote the Legendre-Fenchel transforms, or convex conjugates, of F and H defined by

$$F^*(B^T\lambda) = \sup_{z \in \mathbb{R}^n} \left(\langle B^T\lambda, z \rangle - F(z)\right),$$

$$H^*(A^T\lambda) = \sup_{u \in \mathbb{R}^m} \left(\langle A^T\lambda, u \rangle - H(u)\right).$$

2.2.4 Maximally Decoupled Case

An interesting special case of (P0), which will arise in many of the following examples, is when $H(u) = 0$. This corresponds to

$$\begin{aligned}
&\min && F(z). \\
&u \in \mathbb{R}^m, z \in \mathbb{R}^n \\
&Bz + Au = b
\end{aligned} \tag{P1}$$

As before, the dual functional is given by

$$q_1(\lambda) = -F^*(B^T\lambda) - H^*(A^T\lambda) + \langle \lambda, b \rangle,$$

except here $H^*(A^T\lambda)$ can be interpreted as an indicator function defined by

$$H^*(A^T\lambda) = \begin{cases} 0 & \text{if } A^T\lambda = 0, \\ \infty & \text{otherwise.} \end{cases}$$

This can be interpreted as the constraint $A^T\lambda = 0$, which is equivalent to $P\lambda = \lambda$, where P is the projection onto $\mathrm{Im}(A)^\perp$ defined by

$$P = I - AA^\dagger.$$

Therefore the dual problem for (P1) can be written as

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^d} \quad & -F^*(B^TP\lambda) + \langle P\lambda, b \rangle. \\ A^T\lambda = 0 \end{aligned} \tag{D1}$$

The variable u can also be completely eliminated from the primal problem, which can be equivalently formulated as

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & F(z). \\ P(b - Bz) = 0 \end{aligned} \tag{P2}$$

The associated dual functional is

$$q_2(\lambda) = -F^*(B^TP\lambda) + \langle P\lambda, b \rangle,$$

and the dual problem is therefore

$$\max_{\lambda \in \mathbb{R}^d} -F^*(B^TP\lambda) + \langle P\lambda, b \rangle, \tag{D2}$$

which is identical to (D1) without the constraint. However, since $q_2(\lambda) = q_2(P\lambda)$ the $A^T\lambda = 0$ constraint can be added to (D2) without changing the maximum.

2.3 Algorithms

In this section we start by analyzing Bregman iteration (2.3) applied to (P0) because the first step in deriving the split Bregman algorithm in [GO09] was essentially to take advantage of the separable structure of (2.1) by rewriting it as (P0) and applying Bregman iteration. Then we show an equivalence between ADMM (2.4) and the split Bregman algorithm and present a convergence result by Eckstein and Bertsekas [EB92]. Next we interpret AMA (2.5) and the split inexact Uzawa method (2.7) as modifications of ADMM applied to (P0), and we discuss when they are applicable and why they are useful. Throughout, we also discuss the dual interpretations of Bregman iteration/method of multipliers as gradient ascent, split Bregman/ADMM as Douglas Rachford splitting and AMA as proximal forward backward splitting.

2.3.1 Bregman Iteration and Method of Multipliers

2.3.1.1 Application to Primal Problem

Bregman iteration applied to (P0) yields

Algorithm: Bregman iteration on (P0)

$$
\begin{aligned}
(z^{k+1}, u^{k+1}) = \arg \min_{z\in\mathbb{R}^n, u\in\mathbb{R}^m} & F(z) - F(z^k) - \langle p_z^k, z - z^k\rangle + \\
& H(u) - H(u^k) - \langle p_u^k, u - u^k\rangle + \\
& \frac{\alpha}{2}\|b - Au - Bz\|^2 \\
p_z^{k+1} &= p_z^k + \alpha B^T(b - Au^{k+1} - Bz^{k+1}) \\
p_u^{k+1} &= p_u^k + \alpha A^T(b - Au^{k+1} - Bz^{k+1}).
\end{aligned} \tag{2.12}
$$

For the initialization, p_z^0 and p_u^0 are set to zero while z^0 and u^0 are arbitrary. Note that for $k \geq 1$, $p_u^k \in \partial H(u^k)$ and $p_z^k \in \partial F(z^k)$. Now, following the argument in [YOG08] that shows an equivalence between Bregman iteration and the method of multipliers (2.2) in the case of linear constraints, define λ^k for $k \geq 0$ by $\lambda^0 = 0$ and

$$\lambda^{k+1} = \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}). \tag{2.13}$$

Notice that if $p_z^k = B^T\lambda^k$ and $p_u^k = A^T\lambda^k$ then $p_z^{k+1} = B^T\lambda^{k+1}$ and $p_u^{k+1} = A^T\lambda^{k+1}$. So by induction, it holds for all k. This implies that

$$-\langle p_z^k, z\rangle - \langle p_u^k, u\rangle = -\langle B^T\lambda^k, z\rangle - \langle A^T\lambda^k, u\rangle = \langle \lambda^k, -Au - Bz\rangle.$$

This means the objective function in (2.12) up to a constant is equivalent to the augmented Lagrangian at λ^k, defined by

$$L_\alpha(z, u, \lambda^k) = F(z) + H(u) + \langle \lambda^k, b - Au - Bz\rangle + \frac{\alpha}{2}\|b - Au - Bz\|^2. \tag{2.14}$$

Then (z^{k+1}, u^{k+1}) in (2.12) can be equivalently updated by the method of multipliers (2.2),

---

Algorithm: Method of multipliers on (P0)

$$(z^{k+1}, u^{k+1}) = \arg\min_{z\in\mathbb{R}^n, u\in\mathbb{R}^m} L_\alpha(z, u, \lambda^k) \tag{2.15}$$

$$\lambda^{k+1} = \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}). \tag{2.16}$$

This connection was also pointed out in [TW09].

Note that the same assumptions that guaranteed existence of a minimizer for (P0) also guarantee that (2.15) is well defined. Having assumed that there exists $c \in \mathbb{R}$ such that

$$Q = \{(z, u) : F(z) + H(u) \leq c\ ,\ Au + Bz = b\}$$

is nonempty and bounded, it follows that

$$R = \left\{(z,u) : F(z) + H(u) + \langle \lambda^k, b - Au - Bz\rangle + \frac{\alpha}{2}\|b - Au - Bz\|^2 \leq c\right\}$$

is nonempty and bounded. If not, then being an unbounded convex set, R must contain a half line. Because of the presence of the quadratic term, any such line must be parallel to the affine set defined by $Au + Bz = b$. But since R is also closed, by ([Roc70] 8.3) a half line is also contained in that affine set, which contradicts the assumption that Q was bounded. Weierstrass' theorem can then be used to show that a minimum of (2.15) is attained.

2.3.1.2 Dual Interpretation

Since Bregman iteration with linear constraints is equivalent to the method of multipliers it also shares some of the interesting dual interpretations. In particular, it can be interpreted as a proximal point method for maximizing $q(\lambda)$ or as a gradient ascent method for maximizing $q_\alpha(\lambda)$, where $q_\alpha(\lambda)$ denotes the dual of the augmented Lagrangian L_α defined by

$$q_\alpha(\lambda) = \inf_{z\in\mathbb{R}^n, u\in\mathbb{R}^m} L_\alpha(z,u,\lambda). \tag{2.17}$$

Note that from previous assumptions guaranteeing existence of an optimal solution to (P0), and because the augmented term $\frac{\alpha}{2}\|b - Au - Bz\|^2$ is zero when the constraint is satisfied, the maximums of $q(\lambda)$ and $q_\alpha(\lambda)$ are attained and equal. Following arguments by Rockafellar in [Roc76] and Bertsekas and Tsitsiklis in [BT89], note that

$$L_\alpha(z,u,\lambda^k) = \max_{\lambda\in\mathbb{R}^d} L(z,u,\lambda) - \frac{1}{2\alpha}\|\lambda - \lambda^k\|^2.$$

As in (2.15), let (z^{k+1}, u^{k+1}) (possibly not unique) be where the minimum of $L_\alpha(z,u,\lambda^k)$ is attained. Also let λ^{k+1} be defined as the Lagrange multiplier

update (2.16),

$$\lambda^{k+1} = \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}).$$

We can verify that $(z^{k+1}, u^{k+1}, \lambda^{k+1})$ is a saddle point of $L(z, u, \lambda) - \frac{1}{2\alpha}\|\lambda - \lambda^k\|^2$ by showing that

$$L(z^{k+1}, u^{k+1}, \lambda) - \frac{1}{2\alpha}\|\lambda - \lambda^k\|^2 \leq L(z^{k+1}, u^{k+1}, \lambda^{k+1}) - \frac{1}{2\alpha}\|\lambda^{k+1} - \lambda^k\|^2 \quad (2.18)$$

$$\leq L(z, u, \lambda^{k+1}) - \frac{1}{2\alpha}\|\lambda^{k+1} - \lambda^k\|^2 \quad (2.19)$$

for all (z, u, λ). The first inequality (2.18) is true because

$$\lambda^{k+1} = \arg\max_{\lambda} L(z^{k+1}, u^{k+1}, \lambda) - \frac{1}{2\alpha}\|\lambda - \lambda^k\|^2$$

by definition. For the second inequality, we first notice that by plugging in λ^{k+1},

$$\begin{aligned} &L(z, u, \lambda^{k+1}) - \frac{1}{2\alpha}\|\lambda^{k+1} - \lambda^k\|^2 \\ &= L(z, u, \lambda^k) + \alpha\langle b - Au^{k+1} - Bz^{k+1}, b - Au - Bz\rangle - \frac{\alpha}{2}\|b - Au^{k+1} - Bz^{k+1}\|^2. \end{aligned} \quad (2.20)$$

Furthermore, finding a minimizer of (2.20) is equivalent to solving

$$\arg\min_{z,u} L(z, u, \lambda^k) + \langle \nabla(\frac{\alpha}{2}\|b - Au - Bz\|^2)|_{(z^{k+1}, u^{k+1})}, (z, u)\rangle. \quad (2.21)$$

It follows ([BT89] Lemma 4.1, p. 257) from the fact that (z^{k+1}, u^{k+1}) is a minimizer of $L(z, u, \lambda^k) + \frac{\alpha}{2}\|b - Au - Bz\|^2$ that it is also a minimizer for (2.21). Therefore,

$$(z^{k+1}, u^{k+1}) = \arg\min_{z,u} L(z, u, \lambda^{k+1}) - \frac{1}{2\alpha}\|\lambda^{k+1} - \lambda^k\|^2,$$

verifying the second inequality (2.19).

By the definition of q_α,

$$q_\alpha(\lambda^k) = \min_{z,u}\max_{\lambda} L(z, u, \lambda) - \frac{1}{2\alpha}\|\lambda - \lambda^k\|^2.$$

From the existence of a saddle point, the min and max can be swapped ([Roc70] 36.2), implying

$$q_\alpha(\lambda^k) = \max_\lambda \inf_{z,u} L(z,u,\lambda) - \frac{1}{2\alpha}\|\lambda - \lambda^k\|^2.$$

By the definition of q,

$$q_\alpha(\lambda^k) = \max_\lambda q(\lambda) - \frac{1}{2\alpha}\|\lambda - \lambda^k\|^2, \tag{2.22}$$

and because $(z^{k+1}, u^{k+1}, \lambda^{k+1})$ is a saddle point, this maximum is attained at λ^{k+1} ([Roc70] 36.2). In other words, the Lagrange multiplier update (2.16) is given by

$$\lambda^{k+1} = \arg\max_{\lambda\in\mathbb{R}^d} q(\lambda) - \frac{1}{2\alpha}\|\lambda - \lambda^k\|^2, \tag{2.23}$$

which can be interpreted as a step in a proximal point method for maximizing $q(\lambda)$. The connection to the proximal point method is also derived for example in [BT89]. Since from (2.23), λ^{k+1} is uniquely determined given λ^k, that means that $Au^{k+1} + Bz^{k+1}$ is constant over all minimizers (z^{k+1}, u^{k+1}) of $L_\alpha(z,u,\lambda^k)$. Going back to the Bregman iteration (2.12), we also have that $p_z^{k+1} = B^T\lambda^{k+1}$ and $p_u^{k+1} = A^T\lambda^{k+1}$ were uniquely determined at each iteration.

One way to interpret (2.23) as a gradient ascent method applied to $q_\alpha(\lambda)$ is to note that from (2.22), $q_\alpha(\lambda^k)$ is minus the Moreau envelope of index α of the closed proper convex function $-q$ at λ^k ([CW06] 2.3). The Moreau envelope can be shown to be differentiable, and there is a formula for its gradient ([BT89] p. 234), which when applied to (2.22) yields

$$\nabla q_\alpha(\lambda^k) = -\left[\frac{\lambda^k - \arg\max_\lambda \left(q(\lambda) - \frac{1}{2\alpha}\|\lambda - \lambda^k\|^2\right)}{\alpha}\right].$$

Substituting in λ^{k+1} we see that

$$\nabla q_\alpha(\lambda^k) = \frac{\lambda^{k+1} - \lambda^k}{\alpha},$$

which means we can interpret the Lagrange multiplier update as the gradient ascent step

$$\lambda^{k+1} = \lambda^k + \alpha \nabla q_\alpha(\lambda^k),$$

where $\nabla q_\alpha(\lambda^k) = (b - Au^{k+1} - Bz^{k+1})$.

2.3.2 Split Bregman and ADMM Equivalence

2.3.2.1 Alternating Minimization

The split Bregman algorithm uses an alternating minimization approach to minimize (2.15), namely iterating

$$z^{k+1} = \arg\min_{z \in \mathbb{R}^n} F(z) + \langle \lambda^k, -Bz \rangle + \frac{\alpha}{2} \|b - Au^k - Bz\|^2$$
$$u^{k+1} = \arg\min_{u \in \mathbb{R}^m} H(u) + \langle \lambda^k, -Au \rangle + \frac{\alpha}{2} \|b - Au - Bz^{k+1}\|^2$$

T times and then updating

$$\lambda^{k+1} = \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}).$$

When $T = 1$, this becomes ADMM (2.4), which can be interpreted as alternately minimizing the augmented Lagrangian $L_\alpha(z, u, \lambda)$ with respect to z, then u and then updating the Lagrange multiplier λ,

---

Algorithm: ADMM on (P0)

$$z^{k+1} = \arg\min_{z \in \mathbb{R}^n} F(z) + \langle \lambda^k, -Bz \rangle + \frac{\alpha}{2} \|b - Au^k - Bz\|^2 \tag{2.24a}$$
$$u^{k+1} = \arg\min_{u \in \mathbb{R}^m} H(u) + \langle \lambda^k, -Au \rangle + \frac{\alpha}{2} \|b - Au - Bz^{k+1}\|^2 \tag{2.24b}$$
$$\lambda^{k+1} = \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}). \tag{2.24c}$$

A similar derivation motivated by the augmented Lagrangian can be found in [BT89]. Note that this equivalence between split Bregman and ADMM is not in general true when the constraints are not linear.

Also note the asymmetry of the u and z updates. If we switch the order, first minimizing over u, then over z, we obtain a valid but different incarnation of ADMM, which we are not considering here.

2.3.2.2 Dual Interpretation

Some additional insight comes from the dual interpretation of ADMM as Douglas-Rachford [DR56] splitting applied to the dual problem (D0), which we recall can be written as

$$\max_{\lambda \in \mathbb{R}^d} -F^*(B^T\lambda) + \langle \lambda, b \rangle - H^*(A^T\lambda).$$

Define operators Ψ and ϕ by

$$\Psi(\lambda) = B\partial F^*(B^T\lambda) - b \tag{2.25}$$

$$\phi(\lambda) = A\partial H^*(A^T\lambda). \tag{2.26}$$

Douglas Rachford splitting is a classical method for solving parabolic problems of the form

$$\frac{d\lambda}{dt} + f(\lambda) + g(\lambda) = 0$$

by iterating

$$\begin{aligned} \frac{\hat{\lambda}^k - \lambda^k}{\Delta t} + f(\hat{\lambda}^k) + g(\lambda^k) &= 0 \\ \frac{\lambda^{k+1} - \lambda^k}{\Delta t} + f(\hat{\lambda}^k) + g(\lambda^{k+1}) &= 0, \end{aligned}$$

where Δt is the time step. By iterating to steady state, this can also be used to solve

$$f(\lambda) + g(\lambda) = 0.$$

Solving the dual problem (D0) is equivalent to finding λ such that zero is in the subdifferential of $-q$ at λ. One approach is to look for λ such that

$$0 \in \Psi(\lambda) + \phi(\lambda). \tag{2.27}$$

Such a λ necessarily solves (D0). Some additional minor technical assumptions, usually true in practice, are needed for (2.27) to be equivalent to (D0). See ([Roc70] 23.8, 23.9).

By formally applying Douglas Rachford splitting to (2.27) with α as the time step, we get

$$0 \in \frac{\hat{\lambda}^k - \lambda^k}{\alpha} + \Psi(\hat{\lambda}^k) + \phi(\lambda^k), \tag{2.28a}$$

$$0 \in \frac{\lambda^{k+1} - \lambda^k}{\alpha} + \Psi(\hat{\lambda}^k) + \phi(\lambda^{k+1}). \tag{2.28b}$$

Following the arguments by Glowinski and Le Tallec [GT89] and Eckstein and Bertsekas [EB92], we can show that ADMM satisfies (2.28). Define

$$\hat{\lambda}^k = \lambda^k + \alpha(b - Bz^{k+1} - Au^k).$$

Then from the optimality condition for (2.24a),

$$B^T\hat{\lambda}^k \in \partial F(z^{k+1}).$$

Then from the definitions of subgradient and convex conjugate it follows that

$$z^{k+1} \in \partial F^*(B^T\hat{\lambda}^k).$$

Multiplying by B and subtracting b we have

$$Bz^{k+1} - b \in B\partial F^*(B^T\hat{\lambda}^k) - b = \Psi(\hat{\lambda}^k).$$

The analogous argument starting with the optimality condition for (2.24b) yields

$$Au^{k+1} \in A\partial H^*(A^T\lambda^{k+1}) = \phi(\lambda^{k+1}).$$

With λ^{k+1} defined by (2.24c) and noting that $Au^k \in \phi(\lambda^k)$, we see that the ADMM procedure satisfies (2.28).

It's important to note that Ψ and ϕ are not necessarily single valued, so there could possibly be multiple ways of formally satisfying the Douglas Rachford splitting as written in (2.28). For example, in the maximally decoupled case where $H(u) = 0$, ϕ can be defined by

$$\phi(y) = \begin{cases} \mathrm{Im}(A) & \text{for } y \text{ such that } A^T y = 0 \\ \emptyset & \text{otherwise} \end{cases}.$$

The method of multipliers applied to either (P1) or (P2) with $P\lambda^0 = \lambda^0$ is equivalent to the proximal point method applied to the dual. This would yield

$$\lambda^{k+1} = \hat{\lambda}^k = \arg\max_{y \in \mathbb{R}^d} -F^*(B^T P y) + \langle Py, b\rangle - \frac{1}{2\alpha}\|y - \lambda^k\|^2$$

with $P\lambda^k = \lambda^k$. This also formally satisfies (2.28), but the λ^{k+1} updates are different from ADMM and ususally more difficult to compute.

The particular way in which ADMM satisfies (2.28) can be derived by applying the Moreau decomposition [Mor65, CW06] to directly rewrite ADMM applied to (P0) as Douglas Rachford splitting applied to (D0).

Theorem 2.3.1. *(Generalized Moreau Decomposition)*
If J is a closed proper convex function on $\mathbb{R}^n$, $f \in \mathbb{R}^m$ and $A \in \mathbb{R}^{n\times m}$, then

$$f = \arg\min_u J(Au) + \frac{1}{2\alpha}\|u - f\|_2^2 + \alpha A^T \arg\min_p J^*(p) + \frac{\alpha}{2}\|A^T p - \frac{f}{\alpha}\|_2^2. \quad (2.29)$$

Proof. [Tse09] Let p^* be a minimizer of $J^*(p) + \frac{\alpha}{2}\|A^T p - \frac{f}{\alpha}\|_2^2$. Then

$$0 \in \partial J^*(p^*) + \alpha A(A^T p^* - \frac{f}{\alpha}).$$

Let

$$u^* = f - \alpha A^T p^*.$$

Multiplying by A we see that

$$Au^* \in \partial J^*(p^*).$$

By the definitions of the subdifferential and Legendre transform, this implies

$$\begin{aligned} p^* &\in \partial J(Au^*) \\ A^T p^* &\in A^T \partial J(Au^*) \\ \frac{f-u^*}{\alpha} &\in A^T \partial J(Au^*) \\ 0 &\in A^T \partial J(Au^*) + \frac{u^*-f}{\alpha}. \end{aligned}$$

This implies that

$$u^* = \arg\min_u J(Au) + \frac{1}{2\alpha}\|u-f\|^2,$$

which verifies that the Moreau decomposition is given by

$$f = u^* + \alpha A^T p^*.$$

□

To rewrite ADMM as Douglas Rachford splitting, first combine (2.24b) and (2.24c) from (2.24) to get

$$\lambda^{k+1} = \lambda^k + \alpha(b - Bz^{k+1}) - \alpha A\left[\arg\min_u H(u) + \frac{\alpha}{2}\|Au - \frac{\lambda^k + \alpha(b-Bz^{k+1})}{\alpha}\|^2\right].$$

Applying the Moreau decomposition (2.29) then yields

$$\lambda^{k+1} = \arg\min_\lambda H^*(A^T\lambda) + \frac{1}{2\alpha}\|\lambda - (\lambda^k + \alpha(b - Bz^{k+1}))\|^2.$$

We can also apply the Moreau decomposition to (2.24a) to get

$$\alpha B^T z^{k+1} = \lambda^k + \alpha(b - Au^k) - \arg\min_{\hat\lambda} F^*(B^T\hat\lambda) + \frac{1}{2\alpha}\|\hat\lambda - (\lambda^k + \alpha(b - Au^k))\|^2.$$

Let

$$\hat{\lambda}^k = \arg\min_{\hat{\lambda}} F^*(B^T\hat{\lambda}) - \langle \hat{\lambda}, b\rangle + \frac{1}{2\alpha}\|\hat{\lambda} - (\lambda^k - \alpha A u^k)\|^2. \tag{2.30}$$

Then since $\alpha B^T z^{k+1} = \lambda^k + \alpha(b - Au^k) - \hat{\lambda}^k$,

$$\lambda^{k+1} = \arg\min_{\lambda} H^*(A^T\lambda) + \frac{1}{2\alpha}\|\lambda - \alpha A u^k - \hat{\lambda}^k\|^2. \tag{2.31}$$

It's straightforward to verify that since $Au^k \in \phi(\lambda^k)$ and $Bz^{k+1} - b \in \Psi(\hat{\lambda}^k)$ that (2.30) and (2.31) are consistent with (2.28). These Douglas Rachford steps can furthermore be rewritten in a more implementable way by removing the dependence on Au^k. We can plug the expression for Bz^{k+1} into the Lagrange multiplier update (2.24c), which implies

$$Au^{k+1} = Au^k + \frac{1}{\alpha}(\hat{\lambda}^k - \lambda^{k+1}).$$

Letting $y^k = \lambda^k + \alpha A u^k$, this becomes

$$y^{k+1} = y^k + (\hat{\lambda}^k - \lambda^k).$$

Substituting $Au^k = \frac{y^k - \lambda^k}{\alpha}$ into (2.30) and (2.31) and combining these steps with the y^{k+1} update, we arrive at an implementable form of Douglas Rachford splitting applied to (D0) which produces the same sequence of λ^k as ADMM applied to (P0).

Algorithm: Douglas Rachford on (D0)

$$\hat{\lambda}^k = \arg\min_{\hat{\lambda}} F^*(B^T\hat{\lambda}) - \langle \hat{\lambda}, b\rangle + \frac{1}{2\alpha}\|\hat{\lambda} - (2\lambda^k - y^k)\|^2 \tag{2.32a}$$

$$\lambda^{k+1} = \arg\min_{\lambda} H^*(A^T\lambda) + \frac{1}{2\alpha}\|\lambda - (y^k - \lambda^k + \hat{\lambda}^k)\|^2 \tag{2.32b}$$

$$y^{k+1} = y^k + \hat{\lambda}^k - \lambda^k \tag{2.32c}$$

The following theorem from [Eck89] shows that convergence of λ^k can be ensured with very few assumptions.

Theorem 2.3.2. *[Eck89] Assume F and H are closed proper convex functions. Let $\alpha > 0$ and let (λ^0, y^0) be arbitrary. Suppose $(\hat{\lambda}^k, \lambda^k, y^k)$ satisfies (2.32). Then $\{\lambda^k\}$ converges to a solution of (D0).*

It's also possible to express (2.30) and (2.31) in terms of the resolvents $(I + \alpha\Psi)^{-1}$ and $(I + \alpha\phi)^{-1}$,

$$\hat{\lambda}^k = (I + \alpha\Psi)^{-1}(\lambda^k - \alpha A u^k) \tag{2.33a}$$

$$\lambda^{k+1} = (I + \alpha\phi)^{-1}(\hat{\lambda}^k + \alpha A u^k). \tag{2.33b}$$

Since u^k by assumption is uniquely determined, Au^k is well defined. One way to argue the resolvents are well defined is using monotone operator theory [Eck89]. Briefly, a multivalued operator $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ is monotone if

$$\langle w - w', u - u' \rangle \geq 0 \text{ whenever } w \in \Phi(u) \ , \ w' \in \Phi(u') \ .$$

The operator Φ is maximal monotone if in addition to being monotone, its graph $\{(u, w) \in \mathbb{R}^d \times \mathbb{R}^d | w \in \Phi(u)\}$ is not strictly contained in the graph for any other monotone operator. From a result by Minty [Min62], if Φ is maximal monotone, then for any $\alpha > 0$, $(I + \alpha\Phi)^{-1}$ is single valued and defined on all of $\mathbb{R}^d$ ([EB92], [Tse91]). Then from a result by Rockafellar ([Roc70] 31.5.2), Φ is maximal monotone if it is the subdifferential of a closed proper convex function. Since $\Psi(y)$ and $\phi(y)$ were defined to be subdifferentials of $F^*(B^T y) - \langle y, b \rangle$ and $H^*(A^T y)$ respectively, the resolvents in (2.33) are well defined.

It's possible to rewrite the updates in (2.33) completely in terms of the dual variable [EB92]. Combining the two steps yields

$$\lambda^{k+1} = (I + \alpha\phi)^{-1} \left((I + \alpha\Psi)^{-1}(\lambda^k - \alpha A u^k) + \alpha A u^k \right). \tag{2.34}$$

Suppose

$$y^k = \lambda^k + \alpha A u^k.$$

Since $Au^k \in \phi(\lambda^k)$, $y^k \in (I + \alpha\phi)\lambda^k$. So $\lambda^k = (I + \alpha\phi)^{-1}y^k$. We can use this to rewrite (2.34) as

$$\lambda^{k+1} = (I + \alpha\phi)^{-1}\left[(I + \alpha\Psi)^{-1}\left(2(I + \alpha\phi)^{-1} - I\right) + \left(I - (I + \alpha\phi)^{-1}\right)\right] y^k.$$

Now let

$$y^{k+1} = \left[(I + \alpha\Psi)^{-1}\left(2(I + \alpha\phi)^{-1} - I\right) + \left(I - (I + \alpha\phi)^{-1}\right)\right] y^k. \tag{2.35}$$

Recalling the definition of $\hat{\lambda}^k$ and λ^{k+1}

$$\begin{aligned} y^{k+1} &= \left((I + \alpha\Psi)^{-1}(\lambda^k - \alpha Au^k) + \alpha Au^k\right) \\ &= \hat{\lambda}^k + \alpha Au^k \\ &= \lambda^k + \alpha(b - Bz^{k+1}) \\ &= \lambda^{k+1} + \alpha Au^{k+1}. \end{aligned}$$

Thus assuming we initialize $y^0 = \lambda^0 + \alpha Au^0$ with $u^0 \in \partial H^*(A^T\lambda^0)$, $y^k = \lambda^k + \alpha Au^k$ and $\lambda^k = (I + \alpha\phi)^{-1}y^k$ hold for all $k \geq 0$. So ADMM is equivalent to iterating (2.35). This is the representation used by Eckstein and Bertsekas [EB92] and referred to as the Douglas Rachford recursion. Note that in the maximally decoupled case, $(I + \alpha\phi)^{-1}$ reduces to the projection matrix P, which projects onto $\mathrm{Im}(A)^{\perp}$.

2.3.2.3 Convergence Theory for ADMM

In [EB92], Eckstein and Bertsekas use the dual Douglas Rachford recursion form of ADMM to show that it can be interpreted as an application of the proximal point algorithm. They use this observation to prove a convergence result for ADMM that allows for approximate computation of z^{k+1} and u^{k+1}, as well some over or under relaxation. Their theorem as stated applies to (P0) in the case

when $A = I$, $b = 0$ and B is an arbitrary full column rank matrix, but the same result also holds under slightly weaker assumptions. In particular, we will assume $F(z) + \|Bz\|^2$ and $H(u) + \|Au\|^2$ are strictly convex and let b be nonzero. Note the strict convexity assumptions automatically hold when A and B have full column rank. We restate their result as it applies to (P0) under the slightly weaker assumptions and in the case without over or under relaxation factors.

Theorem 2.3.3. *(Eckstein, Bertsekas [EB92]) Consider the problem (P0) where F and H are closed proper convex functions, $F(z) + \|Bz\|^2$ is strictly convex and $H(u) + \|Au\|^2$ is strictly convex. Let $\lambda^0 \in \mathbb{R}^d$ and $u^0 \in \mathbb{R}^m$ be arbitrary and let $\alpha > 0$. Suppose we are also given sequences $\{\mu_k\}$ and $\{\nu_k\}$ such that $\mu_k \geq 0$, $\nu_k \geq 0$, $\sum_{k=0}^{\infty} \mu_k < \infty$ and $\sum_{k=0}^{\infty} \nu_k < \infty$. Suppose that*

$$\|z^{k+1} - \arg\min_{z \in \mathbb{R}^n} F(z) + \langle \lambda^k, -Bz \rangle + \frac{\alpha}{2}\|b - Au^k - Bz\|^2\| \leq \mu_k \tag{2.36}$$

$$\|u^{k+1} - \arg\min_{u \in \mathbb{R}^m} H(u) + \langle \lambda^k, -Au \rangle + \frac{\alpha}{2}\|b - Au - Bz^{k+1}\|^2\| \leq \nu_k \tag{2.37}$$

$$\lambda^{k+1} = \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}). \tag{2.38}$$

If there exists a saddle point of $L(z, u, \lambda)$ (2.8), then $z^k \to z^*$, $u^k \to u^*$ and $\lambda^k \to \lambda^*$, where (z^*, u^*, λ^*) is such a saddle point. On the other hand, if no such saddle point exists, then at least one of the sequences $\{u^k\}$ or $\{\lambda^k\}$ must be unbounded.

Note that the convergence result carries over to the split Bregman algorithm in the case when the constraints are linear and when only one inner iteration is used.

Only a few minor changes to the proof in [EB92] are needed to accommodate the slightly weaker assumptions made here. The proof that λ^k converges to a solution of the dual problem (D0) remains unchanged. It follows from the equivalence between ADMM applied to (P0) and Douglas Rachford splitting applied

to (D0) and a convergence proof for a generalized form of Douglas Rachford splitting ([EB92] p. 307). The argument that (z^k, u^k, λ^k) converges to a saddle point is what requires the additional assumptions. This is needed to ensure that u^k converges to a solution of (P0). In [EB92] it is assumed that A and B have full column rank, an assumption that doesn't hold for some important image processing models like the TV-l_2 minimization example discussed in Section 2.4.5. In that case, one of the matrices corresponds to the discrete gradient, which doesn't have full column rank. But it can still be true that $F(z) + \|Bz\|^2$ and $H(u) + \|Au\|^2$ are strictly convex, which still ensures the z^{k+1} and u^{k+1} updates are uniquely determined and is enough to guarantee that (z^k, u^k, λ^k) converges to a saddle point. Although the assumptions on A and B have been slightly weakened in Theorem 2.3.3, this version is less general in other ways because it ignores the relaxation factors ρ_k in [EB92], which here we take to be one.

Proof. This proof of theorem 2.3.3 is due to Eckstein and Bertsekas and is taken from their paper [EB92]. The entire proof is not reproduced here. Just enough is sketched to make the changes clear.

Let $J_{\alpha\Psi}$ and $J_{\alpha\phi}$ be shorthand notation for the resolvents $(I + \alpha\Psi)^{-1}$ and $(I + \alpha\phi)^{-1}$ respectively. Also define

$$
\begin{aligned}
y^k &= \lambda^k + \alpha A u^k \qquad , k \geq 0 \\
\hat{\lambda}^k &= \lambda^k + \alpha(b - Bz^{z+1} - Au^k), \qquad k \geq 0 \\
a_k &= \alpha\|B\|\mu_k \qquad , k \geq 0 \\
\beta_0 &= \|\lambda^0 - J_{\alpha\phi}(\lambda^0 - \alpha A u^0)\| \\
\beta_k &= \alpha\|A\|\nu_k, \qquad k \geq 1
\end{aligned}
$$

The main outline of Eckstein and Bertsekas' proof is to first show that

$$(Y1) \qquad \|\lambda^k - J_{\alpha\phi}(y^k)\| \leq \beta_k$$

$$(Y2) \qquad \|\hat{\lambda}^k - J_{\alpha\Psi}(2\lambda^k - y^k)\| \leq a_k$$

$$(Y3) \qquad y^{k+1} = y^k + \hat{\lambda}^k - \lambda^k$$

hold for all $k \geq 0$. If $\beta_k = 0$ and $a_k = 0$ then this would be exactly the form of the Douglas Rachford splitting algorithm in (2.32). To see this, note that since $\lambda^k = (I + \alpha\phi)^{-1} y^k$ (2.32b) can be replaced by

$$\lambda^k = \arg\min_{\lambda} H^*(A^T\lambda) + \frac{1}{2\alpha}\|\lambda - y^k\|^2,$$

and then (2.32b) and (2.32a) can be swapped. Assuming there exists a saddle point of $L(z, u, \lambda)$ (2.8), Eckstein and Bertsekas apply an earlier theorem in their paper to say that $\{y^k\}$ converges. This Douglas Rachford convergence argument that allows for errors in the updates is the main part of their proof of the generalized ADMM (2.3.3). But since this theorem still applies here with the slightly different assumptions, there's no need to reproduce the details. Finally they argue that $z^k \to z^*$, $u^k \to u^*$ and $\lambda^k \to \lambda^*$, where (z^*, u^*, λ^*) is a saddle point of $L(z, u, \lambda)$. Some changes are made to this last part.

Noting that $(Y1)$ is true for $k = 0$, they suppose it is true at iteration k and show it follows that $(Y2)$ is true at k. Define

$$\bar{z}^k = \arg\min_{z \in R^n} F(z) + \langle \lambda^k, -Bz \rangle + \frac{\alpha}{2}\|b - Bz - Au^k\|^2$$

and

$$\tilde{\lambda}^k = \lambda^k + \alpha(b - B\bar{z}^k - Au^k).$$

Note that $\bar{z}^k$ is uniquely determined because $F(z) + \|Bz\|^2$ is strictly convex. From the optimality conditions for the $\bar{z}^k$ update, it follows that

$$\bar{z}^k \in \partial F^*(B^T\tilde{\lambda}^k),$$

and therefore that

$$B\bar{z}^k - b \in \Psi(\tilde{\lambda}^k).$$

Since

$$\tilde{\lambda}^k + \alpha(B\bar{z}^k - b) = \lambda^k - \alpha A u^k \in \tilde{\lambda}^k + \alpha\Psi(\tilde{\lambda}^k),$$

it follows that

$$\tilde{\lambda}^k = J_{\alpha\Psi}(\lambda^k - \alpha A u^k) = J_{\alpha\Psi}(2\lambda^k - y^k).$$

Then

$$\begin{aligned}
&\|\hat{\lambda}^k - J_{\alpha\Psi}(2\lambda^k - y^k)\| = \|\hat{\lambda}^k - \tilde{\lambda}^k\| = \alpha\|B(z^{k+1} - \bar{z}^k)\| \\
&\leq \alpha\|B\|\|z^{k+1} - \bar{z}^k\| \leq \alpha\|B\|\mu_k = a_k.
\end{aligned}$$

Thus $(Y2)$ holds at iteration k. Next they assume $(Y1)$ and $(Y2)$ hold at k and define

$$\begin{aligned}
s^k &= y^k + \hat{\lambda}^k - \lambda^k \\
&= \lambda^k + \alpha(b - Bz^{k+1}) \\
\bar{u}^k &= \arg\min_{u\in\mathbb{R}^m} H(u) + \langle\lambda^k, -Au\rangle + \frac{\alpha}{2}\|b - Bz^{k+1} - Au\|^2 \\
\tilde{s}^k &= \lambda^k + \alpha(b - Bz^{k+1} - A\bar{u}^k).
\end{aligned}$$

$(Y3)$ holds trivially since

$$y^{k+1} = \lambda^{k+1} + \alpha A u^{k+1} = \lambda^k + \alpha(b - Bz^{k+1}) = y^k + \hat{\lambda}^k - \lambda^k.$$

Next, from the assumption that $H(u) + \|Au\|^2$ is strictly convex, it follows that $\bar{u}^k$ is uniquely determined. The optimality condition for the $\bar{u}^k$ update yields

$$\bar{u}^k \in \partial H^*(A^T\tilde{s}^k)$$

from which it follows that

$$A\bar{u}^k \in \phi(\tilde{s}^k).$$

Since

$$s^k = \tilde{s}^k + \alpha A\bar{u}^k \in \tilde{s}^k + \alpha\phi(\tilde{s}^k),$$

we have that

$$\tilde{s}^k = J_{\alpha\phi}(s^k).$$

Noting that $y^{k+1} = s^k$,

$$\|\lambda^{k+1} - J_{\alpha\phi}(y^{k+1})\| = \|\lambda^{k+1} - J_{\alpha\phi}(s^k)\| = \|\lambda^{k+1} - \tilde{s}^k\| = \alpha\|A(u^{k+1} - \bar{u}^k)\|$$
$$\leq \alpha\|A\|\nu_k = \beta_k,$$

which means $(Y1)$ holds at $k+1$. By induction, $(Y1)$, $(Y2)$ and $(Y3)$ hold for all k. Moreover, the sequences $\{\beta_k\}$ and $\{a_k\}$ are summable by definition. Taken together this satisfies the requirements of a previous theorem in ([EB92] p. 307), Theorem 7. If there exists a saddle point $L(z, u, \lambda)$, then in particular there exists an optimal dual solution, in which case Theorem 7 implies that y^k converges to $y^* = \lambda^* + \alpha w^*$ such that $w^* \in \phi(\lambda^*)$ and $-w^* \in \Psi(\lambda^*)$. If there is no saddle point, Theorem 7 implies the sequence $\{y^k\}$ is unbounded, which means either $\{\lambda^k\}$ or $\{u^k\}$ is unbounded. In the case where y^k converges, note that

$$y^* \in \lambda^* + \alpha\phi(\lambda^*),$$

so

$$\lambda^* = J_{\alpha\phi}(y^*).$$

From $(Y1)$ and the continuity of $J_{\alpha\phi}$ it follows that $\lambda^k \to \lambda^*$. Let $w^k = Au^k$. Then $w^k = \frac{y^k - \lambda^k}{\alpha}$, which implies $w^k \to \frac{y^* - \lambda^*}{\alpha} = w^*$. If A had full column rank, we could immediately conclude the convergence of $\{u^k\}$. Instead, define $S(u) = H(u) + \frac{\alpha}{2}\|Au\|^2$, which was assumed to be strictly convex. Rewrite the objective functional for the u minimization step

$$H(u) + \langle\lambda^k, -Au\rangle + \frac{\alpha}{2}\|b - Bz^{k+1} - Au\|^2 = S(u) + \langle\lambda^k, -Au\rangle + \frac{\alpha}{2}\|b - Bz^{k+1}\|^2$$
$$+ \alpha\langle b - Bz^{k+1}, -Au\rangle.$$

The optimality condition for $\bar{u}^k$ then implies that

$$\begin{aligned}
&0 \in \partial S(\bar{u}^k) - A^T(\lambda^k + \alpha(b - Bz^{k+1})) \\
&0 \in \partial S(\bar{u}^k) - A^T(\lambda^{k+1} + \alpha A u^{k+1}) \\
&A^T y^{k+1} \in \partial S(\bar{u}^k) \\
&\bar{u}^k \in \partial S^*(A^T y^{k+1}).
\end{aligned}$$

Since S is strictly convex, S^* is continuously differentiable ([Roc70] 26.3), so $\bar{u}^k = \nabla S^*(A^T y^{k+1})$. Since $\|u^{k+1} - \bar{u}^k\| \to 0$, this implies

$$u^k \to \nabla S^*(A^T y^*).$$

Let $u^* = \nabla S^*(A^T y^*)$. Since $Au^k \to w^*$, we have that $Au^* = w^*$. Now since $\lambda^{k+1} - \lambda^k = \alpha(b - Bz^{k+1} - Au^{k+1}) \to 0$, we have that

$$Bz^{k+1} \to b - Au^*.$$

The argument for the convergence of $\{z^k\}$ is analogous to the one made for $\{u^k\}$. Define $T(z) = F(z) + \frac{\alpha}{2}\|Bz\|^2$, which was assumed to be strictly convex. Then rewrite the objective functional for the z minimization step

$$\begin{aligned}
&F(z) + \langle \lambda^k, -Bz\rangle + \frac{\alpha}{2}\|b - Bz - Au^k\|^2 = T(z) + \langle \lambda^k, -Bz\rangle + \frac{\alpha}{2}\|b - Au^k\|^2 \\
&+ \alpha\langle b - Au^k, -Bz\rangle.
\end{aligned}$$

The optimality condition for $\bar{z}^k$ then implies

$$\begin{aligned}
B^T(\lambda^k + \alpha(b - Au^k)) \in \partial T(\bar{z}^k \\
\bar{z}^k = \nabla T^*(B^T(\lambda^k + \alpha(b - Au^k))).
\end{aligned}$$

Since T^* is continuously differentiable, $\lambda^k \to \lambda^*$, $u^k \to u^*$ and $\|z^{k+1} - \bar{z}^k\| \to 0$,

$$z^k \to z^* := \nabla T^*(B^T(\lambda^* + \alpha(b - Au^*)))$$

and

$$Au^* + Bz^* = b.$$

Now note that we also have $\tilde{\lambda}^k \to \lambda^*$, $\tilde{s}^k \to \lambda^*$, $\bar{z}^k \to z^*$ and $\bar{u}^k \to u^*$. Recalling the optimality conditions for the u and z update steps,

$$\bar{z}^k \in \partial F^*(B^T\tilde{\lambda}^k) \qquad \text{and} \qquad \bar{u}^k \in \partial H^*(A^T\tilde{s}^k).$$

Citing a result by Brezis [Br73] regarding limits of maximal monotone operators, it then follows that

$$z^* \in \partial F^*(B^T\lambda^*) \qquad \text{and} \qquad u^* \in \partial H^*(A^T\lambda^*).$$

These together with $Au^* + Bz^* = b$ are exactly the optimality conditions (2.11) for (P0). Thus (z^*, u^*, λ^*) is a saddle point of $L(z, u, \lambda)$.

□

2.3.3 Decoupling Variables

The quadratic penalty terms of the form $\frac{\alpha}{2}\|Ku - f\|^2$ that appear in the ADMM iterations couple the variables in a way that can make the algorithm computationally expensive. If K has special structure, this may not be a problem. For example, K could be diagonal. Or it might be possible to diagonalize K^TK using fast transforms like the FFT or the DCT. Alternatively, the ADMM iterations can be modified to avoid the difficulty caused by the $\|Ku\|^2$ term. In this section we show how AMA (2.5) and the split inexact Uzawa method (2.7) accomplish this by modifying the ADMM iterations in different ways. AMA essentially removes the offending quadratic penalty, while the split inexact Uzawa method is based on the preconditioning idea from BOS, which adds an additional quadratic penalty chosen so that it cancels the $\|Ku\|^2$ term. A strict convexity assumption is required to apply AMA, but not for the split inexact Uzawa approach.

2.3.3.1 AMA Applied to Primal Problem

In order to apply AMA to (P0), either F or H must be strictly convex. Assume for now that $H(u)$ is strictly convex with modulus $\sigma > 0$. The additional strict convexity assumption is needed so that the step of minimizing the non-augmented Lagrangian is well defined.

Recalling the definitions of Ψ and ϕ (2.25), proximal forward backward splitting (PFBS) [LM79, Pas79, CW06] applied to the dual problem (D0) is defined by

$$\lambda^{k+1} = (I + \alpha\Psi)^{-1}(I - \alpha\phi)\lambda^k, \tag{2.39}$$

where λ^0 is arbitrary. Note that $\phi(\lambda^k)$ is single valued because of the strict convexity of $H(u)$. Also, $(I+\alpha\Psi)^{-1}$ is well defined because Ψ is maximal monotone. So (2.39) determines λ^{k+1} uniquely given λ^k.

As Tseng shows in [Tse91], (2.39) is equivalent to

Algorithm: AMA applied to (P0)

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) - \langle A^T\lambda^k, u\rangle \tag{2.40a}$$

$$z^{k+1} = \arg\min_{z\in\mathbb{R}^n} F(z) - \langle B^T\lambda^k, z\rangle + \frac{\alpha}{2}\|b - Au^{k+1} - Bz\|^2 \tag{2.40b}$$

$$\lambda^{k+1} = \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1}). \tag{2.40c}$$

To see the equivalence, note that optimality of u^{k+1} implies $A^T\lambda^k \in \partial H(u^{k+1})$. It follows that

$$Au^{k+1} \in A\partial H^*(A^T\lambda^k) = \phi(\lambda^k).$$

Similarly, optimality of z^{k+1} implies

$$Bz^{k+1} - b \in \Psi(\lambda^{k+1}).$$

Since $\lambda^{k+1} = \lambda^k + \alpha(b - Au^{k+1} - Bz^{k+1})$,

$$0 \in \lambda^{k+1} + \alpha\Psi(\lambda^{k+1}) - \lambda^k + \alpha\phi(\lambda^k),$$

from which (2.39) follows. AMA and PFBS are are discussed in more detail in Chapter 3 with regard to their close connection to PDHG.

Tseng shows that $\{u^k, z^k\}$ converges to a solution of (P0) and $\{\lambda^k\}$ converges to a solution of (D0) if α, which he allows to depend on k, satisfies the time step restriction

$$\epsilon \leq \alpha_k \leq \frac{4\sigma}{\|A\|^2} - \epsilon \tag{2.41}$$

for some $\epsilon \in (0, \frac{2\sigma}{\|A\|^2})$.

2.3.3.2 BOS Applied to Primal Problem

The BOS algorithm applied to (2.1) was interpreted by Zhang, Burger, Bresson and Osher in [ZBB09] as an inexact Uzawa method. It modifies the augmented Lagrangian not by removing the quadratic penalty, but by adding an additional proximal-like penalty chosen so that the $\|Ku\|^2$ term cancels out. It simplifies the minimization step by decoupling the variables coupled by the constraint matrix K, and it doesn't require the functional J to be strictly convex. In a sense it combines the best advantages of Rockafellar's proximal method of multipliers [Roc76] and Daubechies, Defrise and De Mol's surrogate functional technique [DDM04]. Recall that the method of multipliers (2.2) applied to (2.1) requires solving

$$u^{k+1} = \arg\min_{u \in \mathbb{R}^m} J(u) + \langle \lambda^k, f - Ku \rangle + \frac{\alpha}{2}\|f - Ku\|^2.$$

The inexact Uzawa method in [ZBB09] modifies that objective functional by adding the term

$$\frac{1}{2}\langle u - u^k, (\frac{1}{\delta} - \alpha K^T K)(u - u^k)\rangle,$$

where δ is chosen such that $0 < \delta < \frac{1}{\alpha\|K^TK\|}$ in order that $(\frac{1}{\delta} - \alpha K^TK)$ is positive definite. Combining and rewriting terms yields

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} J(u) + \frac{1}{2\delta}\|u - u^k + \alpha\delta K^T(Ku^k - f - \frac{\lambda^k}{\alpha})\|^2.$$

The new penalty keeps u^{k+1} close to a linear approximation of the old penalty evaluated at u^k, and the iteration is simplified because the variables u are no longer coupled together by K. An important example is the case when $J(u) = \|u\|_1$, in which case the decoupled functional can be explicitly minimized by a shrinkage formula discussed in section 2.4.2. In [ZBO09], the algorithm was combined with split Bregman and applied to more complicated problems such as one involving nonlocal total variation regularization.

2.3.4 Split Inexact Uzawa Applied to Primal Problem

Applying the same decoupling trick from BOS to the ADMM iterations means selectively replacing some quadratic penalties of the form $\frac{\alpha}{2}\|Ku - f\|^2$ with their linearized counterparts $\frac{1}{2\delta}\|u - u^k + \alpha\delta K^T(Ku^k - f)\|^2$. An example application to constrained TV minimization is given in section 2.4.7. This is a special case of the more general form of the split inexact Uzawa algorithm ([ZBO09] Algorithm A_1). Let $\|\cdot\|_Q$ be defined by $\|\cdot\|_Q^2 = \langle Q\cdot, \cdot\rangle$ for positive definite Q. The split inexact Uzawa method modifies the ADMM iterations by adding additional quadratic penalties to the minimization steps and also generalizing the Lagrange multiplier update.

Algorithm: Split Inexact Uzawa on (P0)

$$z^{k+1} = \arg\min_{z\in\mathbb{R}^n} F(z) + \langle \lambda^k, -Bz\rangle + \frac{\alpha}{2}\|b - Au^k - Bz\|^2 + \frac{1}{2}\|z - z^k\|^2_{Q_1} \tag{2.42a}$$

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle \lambda^k, -Au\rangle + \frac{\alpha}{2}\|b - Au - Bz^{k+1}\|^2 + \frac{1}{2}\|u - u^k\|^2_{Q_2} \tag{2.42b}$$

$$C\lambda^{k+1} = C\lambda^k + (b - Au^{k+1} - Bz^{k+1}). \tag{2.42c}$$

where Q_1, Q_2 and C are positive definite matrices, and C is such that $0 < \frac{1}{\lambda^C_m} \leq \alpha$ where λ^C_m is the smallest eigenvalue of C. The quadratic penalties can be effectively linearized by letting $Q_1 = \frac{1}{\delta} - \alpha B^T B$ and $Q_2 = \frac{1}{\delta} - \alpha A^T A$, with $\delta > 0$ and $\alpha > 0$ chosen small enough to ensure positive definiteness.

The convergence theory from [ZBO09] for the split inexact Uzawa method is further discussed in Chapter 3 in connection with a variant of the PDHG algorithm.

2.4 Example Applications

Here we give a few examples of how to write several optimization problems from image processing in the form (P0) so that application of ADMM takes advantage of the separable structure of the problems and produces efficient, numerically stable methods. The example problems that follow involve minimizing combinations of the l_1 norm, the square of the l_2 norm, and a discretized version of the total variation seminorm. ADMM applied to these problems often requires solving a Poisson equation or l_1-l_2 minimization. So we first define the discretizations used,

the discrete cosine transform, which can be used for solving the Poisson equations, and also the shrinkage formulas that solve the l_1-l_2 minimization problems.

2.4.1 Notation Regarding Discretizations Used

A straightforward way to define a discretized version of the total variation semi-norm is by

$$\|u\|_{TV} = \sum_{p=1}^{M_r} \sum_{q=1}^{M_c} \sqrt{(D_1^+ u_{p,q})^2 + (D_2^+ u_{p,q})^2} \tag{2.43}$$

for $u \in \mathbb{R}^{M_r \times M_c}$. Here, D_k^+ represents a forward difference in the k^{th} index and we assume Neumann boundary conditions. It will be useful to instead work with vectorized $u \in \mathbb{R}^{M_r M_c}$ and to rewrite $\|u\|_{TV}$. The convention for vectorizing an M_r by M_c matrix will be to associate the (p, q) element of the matrix with the $(q-1)M_r + p$ element of the vector. Consider a graph $G(\mathcal{E}, \mathcal{V})$ defined by an M_r by M_c grid with $\mathcal{V} = \{1, ..., M_r M_c\}$ the set of $m = M_r M_c$ nodes and $\mathcal{E}$ the set of $e = 2M_r M_c - M_r - M_c$ edges. Assume the nodes are indexed so that the node corresponding to element (p, q) is indexed by $(q-1)M_r + p$. The edges, which will correspond to forward differences, can be indexed arbitrarily.

Define $D \in \mathbb{R}^{e \times m}$ to be the edge-node adjacency matrix for this graph. So for a particular edge $\eta \in \mathcal{E}$ with endpoint indices $i, j \in \mathcal{V}$ and $i < j$, we have

$$D_{\eta,k} = \begin{cases} -1 & \text{for } k = i, \\ 1 & \text{for } k = j, \\ 0 & \text{for } k \neq i, j. \end{cases} \tag{2.44}$$

The matrix D is a discretization of the gradient and $-D^T$ is the corresponding discretization of the divergence. The product $-D^T D$ defines the discrete Laplacian $\triangle$ corresponding to Neumann boundary conditions. It is diagonalized by the basis for the discrete cosine transform. Let $\tilde{g} \in \mathbb{R}^{M_r \times M_c}$ denote the discrete

cosine transform of $g \in \mathbb{R}^{M_r \times M_c}$ defined by

$$\tilde{g}_{s,t} = \sum_{p=1}^{M_r} \sum_{q=1}^{M_c} g_{p,q} \cos\left(\frac{\pi}{M_r}(p-\frac{1}{2})s\right) \cos\left(\frac{\pi}{M_c}(q-\frac{1}{2})t\right)$$

Like the fast Fourier transform, this can be computed with $O(M_r M_c \log(M_r M_c))$ complexity. The discrete Laplacian can be computed by

$$\widetilde{(\triangle g)}_{s,t} = \left(2\cos\left(\frac{\pi(s-1)}{M_r}\right) + 2\cos\left(\frac{\pi(t-1)}{M_c}\right) - 4\right)\tilde{g}_{s,t}.$$

Also define $E \in \mathbb{R}^{e \times m}$ such that

$$E_{\eta,k} = \begin{cases} 1 & \text{if } D_{\eta,k} = -1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.45}$$

The matrix E will be used to identify the edges used in each forward difference. Now define a norm on $\mathbb{R}^e$ by

$$\|w\|_E = \sum_{k=1}^{m} \left(\sqrt{E^T(w^2)}\right)_k. \tag{2.46}$$

Note that in this context, the square root and w^2 denote componentwise operations. Another way to interpret $\|w\|_E$ is as the sum of the l_2 norms of vectors w^ν, where $w^\nu = \begin{bmatrix} \vdots \\ w_e \\ \vdots \end{bmatrix}$ for e such that $E_{e,\nu} = 1$. Typically, away from the boundary, w^ν is of the form $w^\nu = \begin{bmatrix} w_{e_1^\nu} \\ w_{e_2^\nu} \end{bmatrix}$, where e_1^ν and e_2^ν are the edges used in the forward difference at node ν. So in terms of w^ν, $\|w\|_E = \sum_{\nu=1}^{m} \|w^\nu\|_2$. The discrete TV seminorm defined above (2.43) can be written in terms of $\|\cdot\|_E$ as

$$\|u\|_{TV} = \|Du\|_E.$$

Use of the matrix E is nonstandard, but also more general. For example, by redefining D and adding edge weights, it can easily be extended to other discretizations and even nonlocal TV. Such weighted graph formulations are discussed in [ELB08].

By definition, the dual norm $\|\cdot\|_{E^*}$ to $\|\cdot\|_E$ is

$$\|x\|_{E^*} = \max_{\|y\|_E \leq 1} \langle x, y \rangle. \tag{2.47}$$

If x^ν is defined analogously to w^ν, then

$$\|x\|_{E^*} = \max_\nu \|x^\nu\|_2.$$

To see this, note that by the Cauchy Schwarz inequality,

$$\max_{\|y\|_E \leq 1} \langle x, y \rangle = \max_{\sum_{\nu=1}^m \|y^\nu\|_2 \leq 1} \sum_{\nu=1}^m \langle x^\nu, y^\nu \rangle \leq \max_\nu \|x^\nu\|_2 = \|x^{\tilde{\nu}}\|_2 \text{ for some } \tilde{\nu}.$$

The the maximum is trivially attained if $\|x^{\tilde{\nu}}\|_2 = 0$ and otherwise the maximum is attained for y such that $y^\nu = \begin{cases} \frac{x^{\tilde{\nu}}}{\|x^{\tilde{\nu}}\|_2} & \text{if } \nu = \tilde{\nu} \\ 0 & \text{otherwise} \end{cases}$. Altogether in terms of the matrix E,

$$\|w\|_E = \|\sqrt{E^T(w^2)}\|_1 \qquad \text{and} \qquad \|x\|_{E^*} = \|\sqrt{E^T(x^2)}\|_\infty.$$

2.4.2 Shrinkage Formulas

When the original functional involves the l_1 norm or the TV seminorm, application of split Bregman or ADMM will result in l_1-l_2 minimization problems that can be explicitly solved by soft thresholding, or shrinkage formulas, which will be defined in this section.

2.4.2.1 Primal Approach

Consider

$$\min_w \sum_i \left(\mu \|w_i\| + \frac{1}{2} \|w_i - f_i\|^2 \right), \tag{2.48}$$

where $w_i, f_i \in \mathbb{R}^{s_i}$, and $\|\cdot\|$ still denotes the l_2 norm. This decouples into separate problems of the form $\min_{w_i} \Theta_i(w_i)$ where

$$\Theta_i(w_i) = \mu \|w_i\| + \frac{1}{2} \|w_i - f_i\|^2. \tag{2.49}$$

Consider the case when $\|f_i\| \leq \mu$. Then

$$\begin{aligned}
\Theta_i(w_i) &= \mu \|w_i\| + \frac{1}{2} \|w_i\|^2 + \frac{1}{2} \|f_i\|^2 - \langle w_i, f_i \rangle \\
&\geq \mu \|w_i\| + \frac{1}{2} \|w_i\|^2 + \frac{1}{2} \|f_i\|^2 - \|w_i\| \|f_i\| \\
&= \frac{1}{2} \|w_i\|^2 + \frac{1}{2} \|f_i\|^2 + \|w_i\| (\mu - \|f_i\|) \\
&\geq \frac{1}{2} \|f_i\|^2 = \Theta_i(0),
\end{aligned}$$

which implies $w_i = 0$ is the minimizer when $\|f_i\| \leq \mu$. In the case where $\|f_i\| > \mu$, let

$$w_i = (\|f_i\| - \mu) \frac{f_i}{\|f_i\|},$$

which is nonzero by assumption. Then Θ is differentiable at w_i and

$$\nabla \Theta(w_i) = \mu \frac{w_i}{\|w_i\|} + w_i - f_i,$$

which equals zero because

$$\frac{w_i}{\|w_i\|} = \frac{f_i}{\|f_i\|}.$$

So altogether, the minimizer of (2.48) is given by

$$w_i = \begin{cases} w_i = (\|f_i\| - \mu) \frac{f_i}{\|f_i\|} & \text{if } \|f_i\| > \mu \\ 0 & \text{otherwise} \end{cases}. \tag{2.50}$$

When $f_\gamma, w_\gamma \in \mathbb{R}$ are the components of $f, w \in \mathbb{R}^m$, $\frac{f_\gamma}{\|f_\gamma\|}$ is just $\operatorname{sign}(f_\gamma)$. Define the scalar shrinkage operator S by

$$S_\mu(f)_\gamma = \begin{cases} f_\gamma - \mu \operatorname{sign}(f_\gamma) & \text{if } |f_\gamma| > \mu \\ 0 & \text{otherwise} \end{cases}, \tag{2.51}$$

where $\gamma = 1, 2, ..., m$. This can be interpreted as solving the minimization problem,

$$S_\mu(f) = \arg\min_{w \in \mathbb{R}^m} \mu\|w\|_1 + \frac{1}{2}\|w - f\|^2. \tag{2.52}$$

The formula (2.50) can be interpreted as $w_i = S_\mu(\|f_i\|)\frac{f_i}{\|f_i\|}$, which is to say scalar shrinkage of $\|f_i\|$ in the direction of f_i. Note also that the problem of minimizing over $w \in \mathbb{R}^e$

$$\mu\|w\|_E + \frac{1}{2}\|w - z\|^2, \tag{2.53}$$

which arises in TV minimization problems, is of the form (2.48). In the notation of the previous section, it can be written as

$$\min_{w \in \mathbb{R}^e} \sum_{k=1}^{m} \left[\mu \left(\sqrt{E^T(w^2)} \right)_k + \frac{1}{2} \left(E^T(w - z)^2 \right)_k \right].$$

Let

$$s = E\sqrt{E^T(z)^2}.$$

Similar to the scalar case, by applying (2.50) for $\gamma = 1, 2, ..., e$ we can define the operator $\tilde{S}_\mu(z)$ that solves (2.53) by

$$\tilde{S}_\mu(z)_\gamma = \begin{cases} z_\gamma - \mu\frac{z_\gamma}{s_\gamma} & \text{if } s_\gamma > \mu \\ 0 & \text{otherwise} \end{cases}. \tag{2.54}$$

2.4.2.2 Dual Approach

The shrinkage formulas in the previous section can also be directly derived using duality by applying the Moreau decomposition (Theorem 2.3.1) to the l_1-l_2 min-

imization problem (2.48). Define $J(w) = \sum_i \|w_i\|$ so that (2.48) can be written as

$$\min_w J(w) + \frac{1}{2\mu}\|w - f\|^2. \tag{2.55}$$

It's straightforward to compute the Legendre transform of J.

$$\begin{aligned} J^*(p) &= \sup_w \langle p, w\rangle - J(w) \\ &= \sum_i \sup_{w_i} \langle p_i, w_i\rangle - \|w_i\| \\ &= \sum_i \sup_{w_i} \|w_i\|(\|p_i\| - 1) \\ &= \begin{cases} 0 & \text{if } \max_i \|p_i\| \le 1 \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

We could also have used the fact that the Legendre transform of a norm is the indicator function for the unit ball in its dual norm. Applying the Moreau decomposition to (2.55) implies that

$$\begin{aligned} \arg\min_w J(w) + \frac{1}{2\mu}\|w - f\|^2 &= f - \mu \arg\min_p J^*(p) + \frac{\mu}{2}\|p - \frac{f}{\mu}\|^2 \\ &= f - \mu \arg \min_{\{p:\max_i \|p_i\| \le 1\}} \frac{\mu}{2}\|p - \frac{f}{\mu}\|^2 \\ &= f - \mu \Pi_{\{p:\max_i \|p_i\| \le 1\}}(\frac{f}{\mu}) \\ &= f - \Pi_{\{p:\max_i \|p_i\| \le \mu\}}(f), \end{aligned}$$

where Π denotes the orthogonal projection onto the given set. In the scalar shrinkage case (2.52),

$$S_\mu(f) = f - \Pi_{\{p:\|p\|_\infty \le \mu\}}(f). \tag{2.56}$$

Similarly, in the TV case (2.53),

$$\tilde{S}_\mu(z) = z - \Pi_{\{p:\|p\|_{E^*} \le \mu\}}(z), \tag{2.57}$$

which agrees with (2.54).

2.4.3 ADMM Applied to Constrained TV Minimization

One of the example applications of split Bregman that was presented in [GO09] is constrained total variation minimization. Here we consider the same example but in the context of applying ADMM to (P0). Consider

$$\begin{aligned} \min_{u \in \mathbb{R}^m} \quad & \|u\|_{TV}, \\ Ku &= f \end{aligned}$$

which can be rewritten using the norm $\|\cdot\|_E$ defined in section 2.4.1 as

$$\begin{aligned} \min_{u \in \mathbb{R}^m} \quad & \|Du\|_E. \\ Ku &= f \end{aligned} \tag{2.58}$$

Writing this in the form of (P0) while taking advantage of the separable structure, we let

$$z = Du \qquad B = \begin{bmatrix} -I \\ 0 \end{bmatrix} \qquad A = \begin{bmatrix} D \\ K \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ f \end{bmatrix}.$$

Now the problem can be written

$$\begin{aligned} \min_{z \in \mathbb{R}^n, u \in \mathbb{R}^m} \quad & \|z\|_E. \\ Bz + Au &= b \end{aligned}$$

We assume that $\ker(D) \bigcap \ker(K) = \{0\}$, or equivalently that $\ker(K)$ does not contain the vector of all ones. This ensures that A has full column rank, so Theorem 2.3.3 can be used to guarantee convergence of ADMM applied to this problem. Introducing a dual variable λ, the augmented Lagrangian is

$$\|z\|_E + \langle \lambda, b - Bz - Au \rangle + \frac{\alpha}{2}\|b - Bz - Au\|^2.$$

Let $\lambda = \begin{bmatrix} p \\ q \end{bmatrix}$ and rewrite the augmented Lagrangian as

$$\|z\|_E + \langle p, z - Du\rangle + \langle q, f - Ku\rangle + \frac{\alpha}{2}\|z - Du\|^2 + \frac{\alpha}{2}\|f - Ku\|^2.$$

Moving linear terms into the quadratic terms, the ADMM iterations are given by

$$\begin{aligned}
z^{k+1} &= \arg\min_z \|z\|_E + \frac{\alpha}{2}\|z - Du^k + \frac{p^k}{\alpha}\|^2 \\
u^{k+1} &= \arg\min_u \frac{\alpha}{2}\|Du - z^{k+1} - \frac{p^k}{\alpha}\|^2 + \frac{\alpha}{2}\|Ku - f - \frac{q^k}{\alpha}\|^2 \\
p^{k+1} &= p^k + \alpha(z^{k+1} - Du^{k+1}) \\
q^{k+1} &= q^k + \alpha(f - Ku^{k+1}),
\end{aligned}$$

where $p^0 = q^0 = 0$, u^0 is arbitrary and $\alpha > 0$. Note that this example corresponds to the maximally decoupled case, in which the u update has the interesting interpretation of enforcing the constraint $A^T\lambda = 0$. Here, since $D^T p^0 + K^T q^0 = 0$ and by the optimality condition for u^{k+1}, it follows that $D^T p^k + K^T q^k = 0$ for all k. In particular, this makes the q^{k+1} update unnecessary. The explicit ADMM steps reduce to

$$\begin{aligned}
z^{k+1} &= \tilde{S}_{\frac{1}{\alpha}}(Du^k - \frac{p^k}{\alpha}) \\
u^{k+1} &= (-\triangle + K^T K)^{-1}\left(D^T z^{k+1} + \frac{D^T p^k}{\alpha} + K^T f + \frac{K^T q^k}{\alpha}\right) \\
&= (-\triangle + K^T K)^{-1}\left(D^T z^{k+1} + K^T f\right) \\
p^{k+1} &= p^k + \alpha(z^{k+1} - Du^{k+1}).
\end{aligned}$$

Since the discrete cosine basis diagonalizes the discrete Laplacian for Neumann boundary conditions, this can be efficiently solved whenever $K^T K$ can be simultaneously diagonalized.

2.4.4 ADMM Applied to TV-l_1

The same decomposition principle applied to constrained TV minimization also applies to the discrete TV-l_1 minimization problem ([CE04], [CEN06]),

$$\min_{u\in\mathbb{R}^m} \|u\|_{TV} + \beta\|Ku - f\|_1,$$

which can be rewritten as

$$\min_{u\in\mathbb{R}^m} \|Du\|_E + \beta\|Ku - f\|_1. \tag{2.59}$$

Writing this in the form of (P0), we let

$$z = \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} Du \\ Ku - f \end{bmatrix} \qquad B = -I \qquad A = \begin{bmatrix} D \\ K \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ f \end{bmatrix}.$$

Again assume that $\ker(D)\bigcap\ker(K) = \{0\}$, or $\ker(K)$ does not contain the vector of all ones. With this assumption, Theorem 2.3.3 again applies. Introducing the dual variable λ, which we decompose into $\lambda = \begin{bmatrix} p \\ q \end{bmatrix}$, the augmented Lagrangian can be written

$$\|w\|_E + \beta\|v\|_1 + \langle p, w - Du\rangle + \langle q, v - Ku + f\rangle + \frac{\alpha}{2}\|w - Du\|^2 + \frac{\alpha}{2}\|v - Ku + f\|^2.$$

Minimizing over z would correspond to simultaneously minimizing over w and v. But no term in the augmented Lagrangian contains both w and v, so it is equivalent to separately minimizing over w and over v.

The ADMM iterations are given by

$$
\begin{aligned}
w^{k+1} &= \arg\min_w \|w\|_E + \frac{\alpha}{2}\|w - Du^k + \frac{p^k}{\alpha}\|^2 \\
v^{k+1} &= \arg\min_v \beta\|v\|_1 + \frac{\alpha}{2}\|v - Ku^k + f + \frac{q^k}{\alpha}\|^2 \\
u^{k+1} &= \arg\min_u \frac{\alpha}{2}\|Du - w^{k+1} - \frac{p^k}{\alpha}\|^2 + \frac{\alpha}{2}\|Ku - v^{k+1} - f - \frac{q^k}{\alpha}\|^2 \\
p^{k+1} &= p^k + \alpha(w^{k+1} - Du^{k+1}) \\
q^{k+1} &= q^k + \alpha(v^{k+1} - Ku^{k+1} + f),
\end{aligned}
$$

where $p^0 = q^0 = 0$, u^0 is arbitrary and $\alpha > 0$. Again, corresponding to the $A^T\lambda = 0$ constraint in the dual problem, since $D^Tp^0 + K^Tq^0 = 0$ and by the optimality condition for u^{k+1}, it follows that $D^Tp^k + K^Tq^k = 0$ for all k. The explicit formulas for w^{k+1}, v^{k+1} and u^{k+1} are given by

$$
\begin{aligned}
w^{k+1} &= \tilde{S}_{\frac{1}{\alpha}}(Du^k - \frac{p^k}{\alpha}) \\
v^{k+1} &= S_{\frac{\beta}{\alpha}}(Ku^k - f - \frac{q^k}{\alpha}) \\
u^{k+1} &= (-\triangle + K^TK)^{-1}\left(D^Tw^{k+1} + \frac{D^Tp^k}{\alpha} + K^T(v^{k+1} + f) + \frac{K^Tq^k}{\alpha}\right) \\
&= (-\triangle + K^TK)^{-1}\left(D^Tw^{k+1} + K^T(v^{k+1} + f)\right).
\end{aligned}
$$

To get a sense of the speed of this algorithm, we let $K = I$ and test it numerically on a synthetic grayscale image similar to one from [CE04]. The intensities range from 0 to 255 and the image is scaled to sizes 64×64, 128×128, 256×256 and 512×512. Let $\beta = .6$, $.3$, $.15$ and $.075$ for the different sizes respectively. Similarly let $\alpha = .02$, $.01$, $.005$ and $.0025$. Let $\hat{u}$ denote u^k at the first iteration $k > 1$ such that $\|u^k - u^{k-1}\|_\infty \leq .5$, $\|Du^k - w^k\|_\infty \leq .5$ and $\|v^k - u^k + f\|_\infty \leq .5$. The original image f and the result $\hat{u}$ are shown in Figure 2.1. The number of iterations required and time to compute on an average PC running a MATLAB implementation are tabulated in Table 2.1.

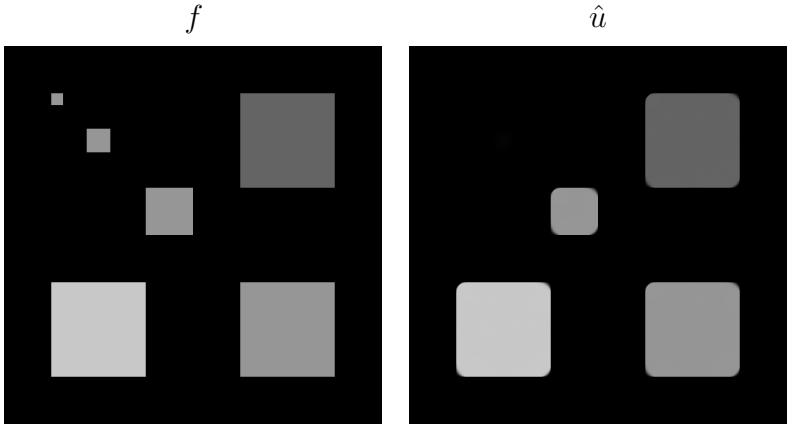


Figure 2.1: TV-l_1 minimization of 512×512 synthetic image

Image Size	Iterations	Time
64×64	40	1s
128×128	51	5s
256×256	136	78s
512×512	359	836s

Table 2.1: Iterations and time required for TV-l_1 minimization

2.4.5 ADMM Applied to TV-l_2

An example where there is more than one effective way to apply ADMM is the TV-l_2 minimization problem

$$\min_{u\in\mathbb{R}^m} \|u\|_{TV} + \frac{\lambda}{2}\|Ku - f\|^2,$$

which can be rewritten as

$$\min_{u\in\mathbb{R}^m} \|Du\|_E + \frac{\lambda}{2}\|Ku - f\|^2. \tag{2.60}$$

The splitting used by Goldstein and Osher for this problem in [GO09] can be written in the form of (P0) by letting

$$z = Du \qquad B = -I \qquad A = D \qquad b = 0.$$

Note that $F(z) = \|z\|_E$ and $H(u) = \frac{\lambda}{2}\|Ku - f\|^2$. Introducing the dual variable p, the augmented Lagrangian can be written

$$\|z\|_E + \frac{\lambda}{2}\|Ku - f\|^2 + \langle p, z - Du\rangle + \frac{\alpha}{2}\|z - Du\|^2.$$

Assume again that $\ker(D)\bigcap\ker(K) = \{0\}$, or $\ker(K)$ does not contain the vector of all ones. This ensures that $\frac{\lambda}{2}\|Ku - f\|^2 + \|Du\|^2$ is strictly convex, so Theorem 2.3.3 applies and guarantees the convergence of ADMM.

The ADMM iterations are given by

$$\begin{aligned} z^{k+1} &= \arg\min_z \|z\|_E + \frac{\alpha}{2}\|z - Du^k + \frac{p^k}{\alpha}\|^2 \\ u^{k+1} &= \arg\min_u \frac{\lambda}{2}\|Ku - f\|^2 + \frac{\alpha}{2}\|Du - z^{k+1} - \frac{p^k}{\alpha}\|^2 \\ p^{k+1} &= p^k + \alpha(z^{k+1} - Du^{k+1}). \end{aligned} \tag{2.61}$$

The explicit formulas for z^{k+1} and u^{k+1} are

$$z^{k+1} = \tilde{S}_{\frac{1}{\alpha}}(Du^k - \frac{p^k}{\alpha})$$
$$u^{k+1} = (-\alpha\triangle + \lambda K^T K)^{-1}\left(\lambda K^T f + \alpha D^T z^{k+1} + D^T p^k\right).$$

Another approach is to apply ADMM to TV-l_2 as it was applied to TV-l_1. This corresponds to the maximally decoupled case and involves adding new variables not just for the TV term but also for the l_2 term when rewriting (2.60) in the form of (P0). Let

$$z = \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} Du \\ Ku - f \end{bmatrix} \qquad B = -I \qquad A = \begin{bmatrix} D \\ K \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ f \end{bmatrix}.$$

Note that $F(z) = \|w\|_E + \frac{\lambda}{2}\|v\|^2$, $H(u) = 0$ and A has full column rank. The augmented Lagrangian can be written

$$\|w\|_E + \frac{\lambda}{2}\|v\|^2 + \langle p, w - Du\rangle + \langle q, v - Ku + f\rangle + \frac{\alpha}{2}\|w - Du\|^2 + \frac{\alpha}{2}\|v - Ku + f\|^2.$$

As with the TV-l_1 example, minimizing over z would correspond to simultaneously minimizing over w and v, which here is equivalent to separately minimizing over w and over v.

The ADMM iterations are then

$$w^{k+1} = \arg\min_w \|w\|_E + \frac{\lambda}{2}\|w - Du^k + \frac{p^k}{\alpha}\|^2$$
$$v^{k+1} = \arg\min_v \frac{\lambda}{2}\|v\|^2 + \frac{\alpha}{2}\|v - Ku^k + f + \frac{q^k}{\alpha}\|^2$$
$$u^{k+1} = \arg\min_u \frac{\alpha}{2}\|Du - w^{k+1} - \frac{p^k}{\alpha}\|^2 + \frac{\alpha}{2}\|Ku - v^{k+1} - f - \frac{q^k}{\alpha}\|^2$$
$$p^{k+1} = p^k + \alpha(w^{k+1} - Du^{k+1})$$
$$q^{k+1} = q^k + \alpha(v^{k+1} - Ku^{k+1} + f).$$

The formulas for w^{k+1}, v^{k+1} and u^{k+1} are

$$\begin{aligned} w^{k+1} &= \tilde{S}_{\frac{1}{\alpha}}(Du^k - \frac{p^k}{\alpha}) \\ v^{k+1} &= \frac{1}{\lambda+\alpha}(\alpha K u^k - \alpha f - q^k) \\ u^{k+1} &= (-\triangle + K^T K)^{-1}\left(K^T f + D^T w^{k+1} + K^T v^{k+1}\right). \end{aligned}$$

By substituting v^{k+1} into the update for u^{k+1} and using the fact that $D^T p^k + K^T q^k = 0$ for all k, the updates for q and v can be eliminated. The remaining iterations are

$$\begin{aligned} w^{k+1} &= \tilde{S}_{\frac{1}{\alpha}}(Du^k - \frac{p^k}{\alpha}) \\ u^{k+1} &= (-\triangle + K^T K)^{-1}\left(\frac{\lambda K^T f}{\lambda+\alpha} + D^T w^{k+1} + \frac{D^T p^k}{\lambda+\alpha} + \frac{\alpha K^T K u^k}{\lambda+\alpha}\right) \\ p^{k+1} &= p^k + \alpha(w^{k+1} - Du^{k+1}). \end{aligned}$$

This alternative application of ADMM to TVL2 is very similar to the first(2.61), differing only in the update for u^{k+1}. Empirically, at least in the denoising case for $K = I$, the two approaches perform similarly. But since the algorithm is neither simplified nor improved by the additional decoupling of the l_2 term, there is no compelling reason to do it.

An approach suggested in [GO09] for speeding up the iterations of (2.61) is to only approximately solve for u^{k+1} using several Gauss Seidel iterations instead of solving a Poisson equation. Convergence of the resulting approximate algorithm could be guaranteed by Theorem 2.3.3 if we knew that the sum of the norms of the errors was finite, but this is a difficult thing to know in advance. Since $H(u)$ was strictly convex in the first method for TV-l_2, an alternative approach to simplifying the iterations is to apply AMA.

2.4.6 AMA Applied to TV-l_2

Consider again the TV-l_2 problem (2.60) in the denoising case where $K = I$. Since $H(u)$ is strictly convex, we can apply AMA to obtain a similar algorithm that doesn't require solving the Poisson equation. Recall the Lagrangian for this problem is given by

$$\|z\|_E + \frac{\lambda}{2}\|u - f\|^2 + \langle p, z - Du\rangle.$$

The AMA iterations are

$$\begin{aligned} u^{k+1} &= \arg\min_u \frac{\lambda}{2}\|u - f\|^2 - \langle D^T p^k, u\rangle \\ z^{k+1} &= \arg\min_z \|z\|_E + \frac{\alpha}{2}\|z - Du^{k+1} + \frac{p^k}{\alpha}\|^2 \qquad (2.62) \\ p^{k+1} &= p^k + \alpha(z^{k+1} - Du^{k+1}). \end{aligned}$$

(2.63)

The explicit formulas for z^{k+1} and u^{k+1} are

$$\begin{aligned} u^{k+1} &= f + \frac{D^T p^k}{\lambda} \\ z^{k+1} &= \tilde{S}_{\frac{1}{\alpha}}(Du^{k+1} - \frac{p^k}{\alpha}). \end{aligned}$$

Note that α must satisfy the time step restriction from (2.41). Since $H(u)$ is strictly convex with modulus $\frac{\lambda}{2}$, a safe choice for α is to let $\alpha \leq \frac{\lambda}{\|D\|^2}$. We can bound $\|D\|^2$ by the largest eigenvalue of $D^T D$, which is minus the discrete Laplacian corresponding to Neumann boundary conditions. The matrix $D^T D$ from its definition has only the numbers 2, 3 and 4 on its main diagonal. All the off diagonal entries are 0 or -1, and the rows sum to zero. Therefore, by the Gersgorin Circle Theorem, all eigenvalues of $D^T D$ are in the interval $[0, 8]$. Thus $\|D\|^2 \leq 8$, so we can take $\alpha = \frac{\lambda}{8}$.

For this example, since it is already efficient to solve the Poisson equation using the discrete cosine transform, the benefit of slightly faster iterations compared to ADMM is outweighed by the reduced stability and the additional iterations required.

The application of AMA to TV-l_2 minimization is equivalent to applying gradient projection (3.37) to its dual problem. This connection will be discussed in greater detail in Section 3.5.

2.4.7 Split Inexact Uzawa Applied to Constrained TV

Consider again the constrained TV minimization problem (2.58) but now with a more complicated matrix K that makes the update for u^{k+1}

$$u^{k+1} = \arg\min_u \frac{\alpha}{2}\|Du - z^{k+1} - \frac{p^k}{\alpha}\|^2 + \frac{\alpha}{2}\|Ku - f - \frac{q^k}{\alpha}\|^2$$

difficult to compute. Applying the split inexact Uzawa algorithm, we can handle the $Ku = f$ constraint in a more explicit manner by adding $\frac{1}{2}\langle u - u^k, (\frac{1}{\delta} - \alpha K^T K)(u - u^k)\rangle$ to the objective functional for the u^{k+1} update, with $0 < \delta < \frac{1}{\alpha\|K^T K\|}$. This yields

$$\begin{aligned} u^{k+1} &= \arg\min_u \frac{\alpha}{2}\|Du - z^{k+1} - \frac{p^k}{\alpha}\|^2 + \frac{1}{2\delta}\|u - u^k + \alpha\delta K^T(Ku^k - f - \frac{q^k}{\alpha})\|^2 \\ &= (\frac{1}{\delta} - \alpha\triangle)^{-1}\left(\alpha D^T z^{k+1} + D^T p^k + \frac{1}{\delta}u^k - \alpha K^T\left(Ku^k - f - \frac{q^k}{\alpha}\right)\right). \end{aligned}$$

Altogether, the modified ADMM iterations are given by

$$\begin{aligned} z^{k+1} &= \tilde{S}_{\frac{1}{\alpha}}(Du^k - \frac{p^k}{\alpha}) \\ u^{k+1} &= (\frac{1}{\delta} - \alpha\triangle)^{-1}\left(\alpha D^T z^{k+1} + D^T p^k + \frac{1}{\delta}u^k - \alpha K^T\left(Ku^k - f - \frac{q^k}{\alpha}\right)\right) \\ p^{k+1} &= p^k + \alpha(z^{k+1} - Du^{k+1}) \\ q^{k+1} &= q^k + \alpha(f - Ku^{k+1}). \end{aligned}$$

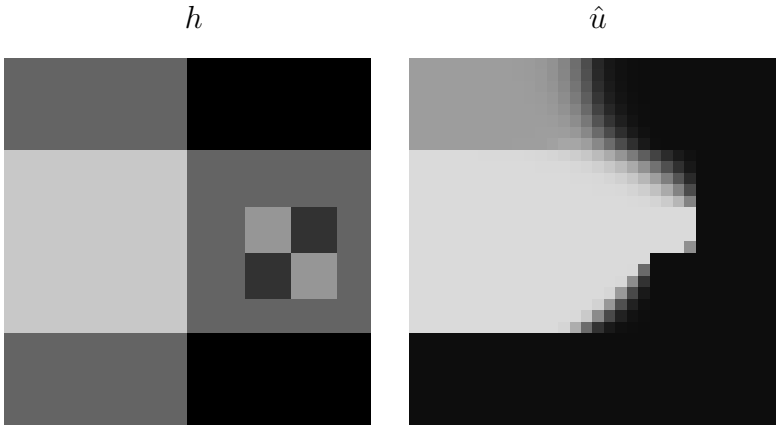


Figure 2.2: Constrained TV minimization of 32×32 image subject to constraints on 4 Haar wavelet coefficients

Although it no longer follows that $D^T p^k + K^T q^k = 0$ as it did for ADMM applied to constrained TV, all updates except for the u^{k+1} step remain the same.

As a numerical test, we will apply this algorithm to a TV wavelet inpainting type problem [CSZ06]. Let $K = \mathcal{X}\psi$, where $\mathcal{X}$ is a row selector and ψ is the matrix corresponding to the translation invariant Haar wavelet transform. For a $2^r \times 2^r$ image, there are $(1 + 3r)2^{2r}$ Haar wavelets when all translations are included. The rows of the $(1 + 3r)2^{2r} \times 2^{2r}$ matrix ψ contain these wavelets weighted such that $\psi^T \psi = I$. $\mathcal{X}$ is a diagonal matrix with ones and zeros on the diagonal. For a simple example, let h be a 32×32 image that is a linear combination of four Haar wavelets. Let $\mathcal{X}$ select the corresponding wavelet coefficients and define $f = \mathcal{X}\psi h$. Also choose $\alpha = .01$ and $\delta = 50$. Let $\hat{u} = u^{10000}$, the result after 10000 iterations. Figure 2.2 shows h and $\hat{u}$. Although $\hat{u}$ may look unusual, it satisfies the four constraints and does indeed have smaller total variation. $\|h\|_{TV} = 1.25 \times 10^4$ whereas $\|\hat{u}\|_{TV} = 1.04 \times 10^4$.

Perhaps a more illustrative example is to try to recover an image from partial

Figure 2.3: Constrained TV minimization of 256×256 cameraman image given 1% of its translation invariant Haar wavelet coeffienents

knowledge of its wavelet coefficients like in the examples of [CSZ06]. Let h be the 256×256 cameraman image shown on the left in Figure 2.3. Let ψ again be the translation invariant Haar wavelet transform as in the previous example. Note that for this size image, these translation invariant Haar wavelets are redundant by a factor of 25. Let $\mathcal{X}$ be a row selector that randomly selects one percent of these wavelet coefficients and define $f = \mathcal{X}\psi h$. Given f, we try to recover h by finding a minimizer u of (2.58). Let $\hat{u} = u^{1500}$ denote the result after 1500 iterations, which is shown on the right in Figure 2.3.

CHAPTER 3

A General Framework for a Class of First Order Primal-Dual Algorithms

3.1 Introduction

In this chapter, we study the primal dual hybrid gradient (PDHG) algorithm proposed by Zhu and Chan [ZC08] and draw connections between it and related algorithms including ADMM, AMA, and the split inexact Uzawa method.

The PDHG method starts with a saddle point formulation of the problem and proceeds by alternating proximal steps that alternately maximize and minimize a penalized form of the saddle function. PDHG can generally be applied to saddle point formulations of inverse problems that can be formulated as minimizing a convex fidelity term plus a convex regularizing term. However, its performance for problems like TV denoising is of special interest since it compares favorably with other popular methods like Chambolle's method [Cha04] and split Bregman [GO09].

PDHG is an example of a first order method, meaning it only requires functional and gradient evaluations. Other examples of first order methods popular for TV minimization include gradient descent, Chambolle's method and split Bregman. Second order methods like the method of Chan, Golub and Mulet (CGM) [CGM99] work by essentially applying Newton's method to an appro-

priate formulation of the Euler Lagrange equations and therefore also require information about the Hessian. These can be quadratically convergent and are useful for computing benchmark solutions of high accuracy. However, the cost per iteration is much higher, so for large scale problems or when high accuracy is not required, these are often less practical than the first order methods that have much lower cost per iteration.

PDHG is also an example of a primal-dual method. Each iteration updates both a primal and a dual variable. It is thus able to avoid some of the difficulties that arise when working only on the primal or dual side. For example, for TV minimization, gradient descent applied to the primal functional has trouble where the gradient of the solution is zero because the functional is not differentiable there. Chambolle's method is a method on the dual that is very effective for TV denoising, but doesn't easily extend to applications where the dual problem is more complicated, such as TV deblurring. Primal-dual algorithms can avoid to some extent these difficulties. Other examples include CGM [CGM99], split Bregman [GO09], and more generally other Bregman iterative algorithms [YOG08] and Lagrangian-based methods.

An adaptive time stepping scheme for PDHG was proposed in [ZC08] and shown to outperform other popular TV denoising algorithms like Chambolle's method, CGM and split Bregman in many numerical experiments with a wide variety of stopping conditions. Aside from some special cases of the PDHG algorithm like gradient projection and subgradient descent, the theoretical convergence properties were not known.

We show that we can make a small modification to the PDHG algorithm, which has little effect on its performance, but that allows the modified algorithm to be interpreted as an inexact Uzawa method of the type analyzed in [ZBO09].

The specific modified PDHG algorithm applied here has been previously proposed by Pock, Cremers, Bischof and Chambolle [PCB09] for minimizing the Mumford-Shah functional. They also prove convergence for a special class of saddle point problems. Here, in a more general setting, we apply the convergence analysis for the inexact Uzawa method from [ZBO09] to show the modified PDHG algorithm converges for a range of fixed parameters. An alternate proof can be found in [CCN09]. While the modified PDHG method is nearly as effective as fixed parameter versions of PDHG, well chosen adaptive step sizes are an improvement. With additional restrictions on the step size parameters, we prove a convergence result for PDHG applied to TV denoising by interpreting it as a projected averaged gradient method on the dual.

We additionally show that the modified PDHG method can be extended in the same ways as PDHG was extended in [ZC08] to apply to additional problems like TV deblurring, l_1 minimization and constrained minimization problems. For these extensions we point out the range of parameters for which the convergence theory from [ZBO09] is applicable. We gain some insight into why the method works by putting it in a general framework and comparing it to related algorithms.

The organization of this chapter is as follows. In Sections 3.2 and 3.3 we review the main idea of the PDHG algorithm and details about its application to TV deblurring type problems. Then in Section 3.4, we discuss primal-dual formulations for a more general problem. We define a general version of PDHG and discuss in detail the framework in which it can be related to other similar algorithms. These connections are diagrammed in Figure 3.1. In Section 3.5 we show how to interpret PDHG applied to TV denoising as a projected averaged gradient method on the dual and present a convergence result for a special case. Then in Section 3.6, we discuss how to use operator splitting to apply the modified

PDHG algorithm to more general problems. In particular, we give examples of its application to constrained TV and l_1 minimization problems and even to multiphase image segmentation. Section 3.7 presents numerical experiments for TV denoising, constrained TV deblurring and constrained l_1 minimization, comparing the performance of the modified PDHG algorithm with other methods.

3.2 Background and Notation

The PDHG algorithm in a general setting is a method for solving problems of the form

$$\min_{u\in\mathbb{R}^m} J(Au) + H(u),$$

where J and H are closed proper convex functions and $A \in \mathbb{R}^{n\times m}$. Usually, $J(Au)$ will correspond to a regularizing term of the form $\|Au\|$, in which case PDHG works by using duality to rewrite it as the saddle point problem

$$\min_{u\in\mathbb{R}^m} \max_{\|p\|_*\leq 1} \langle p, Au\rangle + H(u)$$

and then alternating dual and primal steps of the form

$$p^{k+1} = \arg\max_{\|p\|_*\leq 1} \langle p, Au^k\rangle - \frac{1}{2\delta_k}\|p - p^k\|_2^2$$
$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} \langle p^{k+1}, Au\rangle + H(u) + \frac{1}{2\alpha_k}\|u - u^k\|_2^2$$

for appropriate parameters α_k and δ_k. Here, $\|\cdot\|$ denotes an arbitrary norm on $\mathbb{R}^m$ and $\|\cdot\|_*$ denotes its dual norm defined by

$$\|x\|_* = \max_{\|y\|\leq 1} \langle x, y\rangle,$$

where $\langle\cdot,\cdot\rangle$ is the standard Euclidean inner product. Formulating the saddle point problem also uses the fact that $\|\cdot\|_{**} = \|\cdot\|$ [HJ85], from which it follows that $\|Au\| = \max_{\|p\|_*\leq 1}\langle p, Au\rangle$.

The applications considered here are to solve constrained and unconstrained TV and l_1 minimization problems. Problems of the form

$$\min_{u \in \mathbb{R}^m} \|u\|_{TV} + \frac{\lambda}{2}\|Ku - f\|_2^2 \tag{3.1}$$

are analyzed in [ZC08]. If K is a linear blurring operator, this corresponds to a TV regularized deblurring model. It also includes the TV denoising case when $K = I$. Also mentioned in [ZC08] are possible extensions such as to TV denoising with a constraint on the variance of u and l_1 minimization.

We will continue to use the same notation as in Chapter 2 for the discrete gradient D (2.44), the matrix E (2.45), the norm $\|\cdot\|_E$ (2.46) defined so $\|u\|_{TV} = \|Du\|_E$, and the dual norm $\|\cdot\|_{E^*}$ (2.47).

3.3 PDHG for TV Deblurring

In this section we review from [ZC08] the application of PDHG to the TV deblurring and denoising problems, but using the present notation.

3.3.1 Saddle Point Formulations

For TV minimization problems, the saddle point formulation that the PDHG is based on starts with the observation that

$$\|u\|_{TV} = \max_{p \in X} \langle p, Du \rangle, \tag{3.2}$$

where

$$X = \{p \in \mathbb{R}^e : \|p\|_{E^*} \leq 1\}. \tag{3.3}$$

The set X, which is the unit ball in the dual norm of $\|\cdot\|_E$, can also be interpreted as a Cartesian product of unit balls in the l_2 norm. For example, in order for

Du to be in X, the discretized gradient $\begin{bmatrix} u_{p+1,q} - u_{p,q} \\ u_{p,q+1} - u_{p,q} \end{bmatrix}$ of u at each node (p,q) would have to have Euclidean norm less than or equal to 1. The dual norm interpretation is one way to explain (3.2) since

$$\max_{\|p\|_{E^*} \leq 1} \langle p, Du \rangle = \|Du\|_E,$$

which equals $\|u\|_{TV}$ by definition. Using duality to rewrite $\|u\|_{TV}$ is also the starting point for the primal-dual approach used by CGM [CGM99] and a second order cone programming (SOCP) formulation used in [GY05]. Here it can be used to reformulate problem (3.1) as the min-max problem

$$\min_{u \in \mathbb{R}^m} \max_{p \in X} \Phi(u,p) := \langle p, Du \rangle + \frac{\lambda}{2} \|Ku - f\|_2^2. \tag{3.4}$$

3.3.2 Existence of Saddle Point

One way to ensure that there exists a saddle point (u^*, p^*) of the convex-concave function Φ is to restrict u and p to be in bounded sets. Existence then follows from ([Roc70] 37.6). The dual variable p is already required to lie in the convex set X. Assume that

$$\ker(D) \bigcap \ker(K) = \{0\}.$$

This is equivalent to assuming that $\ker(K)$ does not contain the vector of all ones, which is very reasonable for deblurring problems where K is an averaging operator. With this assumption, it follows that there exists $c \in \mathbb{R}$ such that the set

$$\left\{ u : \|Du\|_E + \frac{\lambda}{2} \|Ku - f\|_2^2 \leq c \right\}$$

is nonempty and bounded. Thus we can restrict u to a bounded convex set.

3.3.3 Optimality Conditions

If (u^*, p^*) is a saddle point of Φ, it follows that

$$\max_{p \in X} \langle p, Du^* \rangle + \frac{\lambda}{2}\|Ku^* - f\|_2^2 = \Phi(u^*, p^*) = \min_{u \in \mathbb{R}^m} \langle p^*, Du \rangle + \frac{\lambda}{2}\|Ku - f\|_2^2,$$

from which we can deduce the optimality conditions

$$D^T p^* + \lambda K^T (Ku^* - f) = 0 \tag{3.5}$$

$$p^* E \sqrt{E^T (Du^*)^2} = Du^* \tag{3.6}$$

$$p^* \in X. \tag{3.7}$$

The second optimality condition (3.6) with E defined by (2.45) can be understood as a discretization of $p^*|\nabla u^*| = \nabla u^*$.

3.3.4 PDHG Algorithm

In [ZC08] it is shown how to interpret the PDHG algorithm applied to (3.1) as a primal-dual proximal point method for solving (3.4) by iterating

$$p^{k+1} = \arg\max_{p \in X} \langle p, Du^k \rangle - \frac{1}{2\lambda\tau_k}\|p - p^k\|_2^2 \tag{3.8a}$$

$$u^{k+1} = \arg\min_{u \in \mathbb{R}^m} \langle p^{k+1}, Du \rangle + \frac{\lambda}{2}\|Ku - f\|_2^2 + \frac{\lambda(1-\theta_k)}{2\theta_k}\|u - u^k\|_2^2. \tag{3.8b}$$

The index k denotes the current iteration. Also, τ_k and θ_k are the dual and primal step sizes respectively. The above max and min problems can be explicitly solved, yielding

Algorithm: PDHG for TV Deblurring

$$p^{k+1} = \Pi_X \left(p^k + \tau_k \lambda D u^k\right) \tag{3.9a}$$

$$u^{k+1} = \left((1-\theta_k) + \theta_k K^T K\right)^{-1} \left((1-\theta_k)u^k + \theta_k (K^T f - \frac{1}{\lambda} D^T p^{k+1})\right). \tag{3.9b}$$

Here, Π_X is the orthogonal projection onto X defined by

$$\Pi_X(q) = \arg\min_{p \in X} \|p - q\|_2^2 = \frac{q}{E \max\left(\sqrt{E^T(q^2)}, 1\right)}, \tag{3.10}$$

where the division and max are understood in a componentwise sense. For example, $\Pi_X(Du)$ can be thought of as a discretization of

$$\begin{cases} \frac{\nabla u}{|\nabla u|} & \text{if } |\nabla u| > 1 \\ \nabla u & \text{otherwise} \end{cases}.$$

This projection Π_X is the same as the one that appeared in the shrinkage formula (2.57). In the denoising case where $K = I$, the p^{k+1} update remains the same and the u^{k+1} simplifies to

$$u^{k+1} = (1-\theta_k)u^k + \theta_k (f - \frac{1}{\lambda} D^T p^{k+1}).$$

For the initialization, we can take $u^0 \in \mathbb{R}^m$ and $p^0 \in X$.

3.4 General Algorithm Framework

In this section we consider a more general class of problems that PDHG can be applied to. We define equivalent primal, dual and several primal-dual formulations. We also place PDHG in a general framework that connects it to other related alternating direction methods applied to saddle point problems.

3.4.1 Primal-Dual Formulations

PDHG can more generally be applied to what we will refer to as the primal problem

$$\min_{u \in \mathbb{R}^m} F_P(u), \tag{P}$$

where

$$F_P(u) = J(Au) + H(u), \tag{3.11}$$

$A \in \mathbb{R}^{n \times m}$, $J : \mathbb{R}^n \to (-\infty, \infty]$ and $H : \mathbb{R}^m \to (-\infty, \infty]$ are closed convex functions. The form of (P) is chosen to be slightly simpler than (P0) in order to facilitate comparisons between related algorithms in this chapter. Assume there exists a solution u^* to (P). We will pay special attention to the case where $J(Au) = \|Au\|$ for some norm $\|\cdot\|$, but this assumption is not required. $J(Au)$ reduces to $\|u\|_{TV}$ when $J = \|\cdot\|_E$ and $A = D$. In Section 3.2 when J was a norm, it was shown how to use the dual norm to define a saddle point formulation of (P) as

$$\min_{u \in \mathbb{R}^m} \max_{\|p\|_* \leq 1} \langle Au, p \rangle + H(u).$$

This can equivalently be written in terms of the Legendre-Fenchel transform, or convex conjugate, of J denoted by J^* and defined by

$$J^*(p) = \sup_{w \in \mathbb{R}^n} \langle p, w \rangle - J(w).$$

When J is a closed proper convex function, we have that $J^{**} = J$ [ET99]. Therefore,

$$J(Au) = \sup_{p \in \mathbb{R}^n} \langle p, Au \rangle - J^*(p).$$

So an equivalent saddle point formulation of (P) is

$$\min_{u \in \mathbb{R}^m} \sup_{p \in \mathbb{R}^n} L_{PD}(u, p), \tag{PD}$$

where

$$L_{PD} = \langle p, Au \rangle - J^*(p) + H(u).$$

This holds even when J is not a norm, but in the case when $J(w) = \|w\|$, we can then use the dual norm representation of $\|w\|$ to write

$$\begin{aligned} J^*(p) &= \sup_w \langle p, w \rangle - \max_{\|y\|_* \leq 1} \langle w, y \rangle \\ &= \begin{cases} 0 & \text{if } \|p\|_* \leq 1 \\ \infty & \text{otherwise} \end{cases}, \end{aligned}$$

in which case we can interpret J^* as the indicator function for the unit ball in the dual norm.

Let (u^*, p^*) be a saddle point of L_{PD}. In particular, this means

$$\max_{p \in \mathbb{R}^n} \langle p, Au^* \rangle - J^*(p) + H(u^*) = L_{PD}(u^*, p^*) = \min_{u \in \mathbb{R}^m} \langle p^*, Au \rangle + H(u) - J^*(p^*),$$

from which we can deduce the equivalent optimality conditions and then use the definitions of the Legendre transform and subdifferential to write these conditions in two ways

$$-A^T p^* \in \partial H(u^*) \qquad \Leftrightarrow \qquad u^* \in \partial H^*(-A^T p^*) \tag{3.12}$$

$$Au^* \in \partial J^*(p^*) \qquad \Leftrightarrow \qquad p^* \in \partial J(Au^*), \tag{3.13}$$

where ∂ denotes the subdifferential. Recall that the subdifferential $\partial F(x)$ of a convex function $F : \mathbb{R}^m \to (-\infty, \infty]$ at the point x is defined by the set

$$\partial F(x) = \{q \in \mathbb{R}^m : F(y) \geq F(x) + \langle q, y - x \rangle \ \forall y \in \mathbb{R}^m\}.$$

Another useful saddle point formulation that we will refer to as the split primal problem is obtained by introducing the constraint $w = Au$ in (P) and forming the Lagrangian

$$L_P(u, w, p) = J(w) + H(u) + \langle p, Au - w \rangle. \tag{3.14}$$

The corresponding saddle point problem is

$$\max_{p\in\mathbb{R}^n} \inf_{u\in\mathbb{R}^m, w\in\mathbb{R}^n} L_P(u,w,p). \tag{$\mathrm{SP_P}$}$$

Although p was introduced in (3.14) as a Lagrange multiplier for the constraint $Au = w$, it has the same interpretation as the dual variable p in (PD). It follows immediately from the optimality conditions that if (u^*, w^*, p^*) is a saddle point for ($\mathrm{SP_P}$), then (u^*, p^*) is a saddle point for (PD).

The dual problem is

$$\max_{p\in\mathbb{R}^n} F_D(p), \tag{D}$$

where the dual functional $F_D(p)$ is a concave function defined by

$$F_D(p) = \inf_{u\in\mathbb{R}^m} L_{PD}(u,p) = \inf_{u\in\mathbb{R}^m} \langle p, Au\rangle - J^*(p) + H(u) = -J^*(p) - H^*(-A^T p). \tag{3.15}$$

Note that this is equivalent to defining the dual by

$$F_D(p) = \inf_{u\in\mathbb{R}^m, w\in\mathbb{R}^n} L_P(u,w,p). \tag{3.16}$$

Since we assumed there exists an optimal solution u^* to the convex problem (P), it follows from Fenchel Duality ([Roc70] 31.2.1) that there exists an optimal solution p^* to (D) and $F_P(u^*) = F_D(p^*)$. Moreover, u^* solves (P) and p^* solves (D) if and only if (u^*, p^*) is a saddle point of $L_{PD}(u,p)$ ([Roc70] 36.2).

By introducing the constraint $y = -A^T p$ in (D) and forming the corresponding Lagrangian

$$L_D(p,y,u) = J^*(p) + H^*(y) + \langle u, -A^T p - y\rangle, \tag{3.17}$$

we obtain yet another saddle point problem,

$$\max_{u\in\mathbb{R}^m} \inf_{p\in\mathbb{R}^n, y\in\mathbb{R}^m} L_D(p,y,u), \tag{$\mathrm{SP_D}$}$$

which we will refer to as the split dual problem. Although u was introduced in ($\mathrm{SP_D}$) as a Lagrange multiplier for the constraint $y = -A^T p$, it actually has the same interpretation as the primal variable u in (P). Again, it follows from the optimality conditions that if (p^*, y^*, u^*) is a saddle point for ($\mathrm{SP_D}$), then (u^*, p^*) is a saddle point for (PD). Note also that

$$F_P(u) = -\inf_{p\in\mathbb{R}^n, y\in\mathbb{R}^m} L_D(p, y, u).$$

3.4.2 Algorithm Framework and Connections to PDHG

In this section we define a general version of PDHG applied to (PD) and discuss connections to related algorithms that can be interpreted as alternating direction methods applied to ($\mathrm{SP_P}$) and ($\mathrm{SP_D}$). These connections are summarized in Figure 3.1.

It was shown in [ZC08] that PDHG applied to TV denoising can be interpreted as a primal-dual proximal point method applied to a saddle point formulation of the problem. More generally, applied to (PD) it yields

---

Algorithm: PDHG on (PD)

$$p^{k+1} = \arg\max_{p\in\mathbb{R}^n} -J^*(p) + \langle p, Au^k\rangle - \frac{1}{2\delta_k}\|p - p^k\|_2^2 \tag{3.18a}$$

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle A^T p^{k+1}, u\rangle + \frac{1}{2\alpha_k}\|u - u^k\|_2^2, \tag{3.18b}$$

---

where $p^0 = 0$, u^0 is arbitrary, and $\alpha_k, \delta_k > 0$. The parameters τ_k and θ_k from (3.9) in terms of δ_k and α_k are

$$\theta_k = \frac{\lambda\alpha_k}{1 + \alpha_k\lambda} \qquad \tau_k = \frac{\delta_k}{\lambda}.$$

3.4.2.1 Proximal Forward Backward Splitting Special Cases of PDHG

Two notable special cases of PDHG are $\alpha_k = \infty$ and $\delta_k = \infty$. These special cases correspond to the proximal forward backward splitting method (PFBS) [LM79, Pas79, CW06] applied to (D) and (P) respectively.

PFBS is an iterative splitting method that can be used to find a minimum of a sum of two convex functionals by alternating a (sub)gradient descent step with a proximal step. Applied to (D) it yields

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \frac{1}{2\delta_k}\|p - (p^k + \delta_k A u^{k+1})\|_2^2, \tag{3.19}$$

where $u^{k+1} \in \partial H^*(-A^T p^k)$. Since $u^{k+1} \in \partial H^*(-A^T p^k) \Leftrightarrow -A^T p^k \in \partial H(u^{k+1})$, which is equivalent to

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle A^T p^k, u\rangle,$$

(3.19) can be written as

Algorithm: PFBS on (D)

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle A^T p^k, u\rangle \tag{3.20a}$$

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \langle p, -A u^{k+1}\rangle + \frac{1}{2\delta_k}\|p - p^k\|_2^2. \tag{3.20b}$$

Even though the order of the updates is reversed relative to PDHG, since the initialization is arbitrary it is still a special case of (3.18) where $\alpha_k = \infty$.

If we assume that $J(\cdot) = \|\cdot\|$, we can interpret the p^{k+1} step as an orthogonal projection onto a convex set,

$$p^{k+1} = \Pi_{\{p:\|p\|_*\leq 1\}}\left(p^k + \delta_k A u^{k+1}\right).$$

Then PFBS applied to (D) can be interpreted as a (sub)gradient projection algorithm.

As a special case of ([CW06] Theorem 3.4), the following convergence result applies to (3.20).

Theorem 3.4.1. *Fix $p^0 \in \mathbb{R}^n$, $u^0 \in \mathbb{R}^m$ and let (u^k, p^k) be defined by (3.20). If H^* is differentiable, $\nabla(H^*(-A^T p))$ is Lipschitz continuous with Lipschitz constant equal to $\frac{1}{\beta}$, and $0 < \inf \delta_k \leq \sup \delta_k < 2\beta$, then $\{p^k\}$ converges to a solution of (D) and $\{u^k\}$ converges to the unique solution of (P).*

Proof. Convergence of $\{p^k\}$ to a solution of (D) follows from ([CW06] 3.4). From (3.20a), u^{k+1} satisfies $-A^T p^k \in \partial H(u^{k+1})$, which, from the definitions of the subdifferential and Legendre transform, implies that $u^{k+1} = \nabla H^*(-A^T p^k)$. So by continuity of ∇H^*, $u^k \to u^* = \nabla H^*(-A^T p^*)$. From (3.20b) and the convergence of $\{p^k\}$, $Au^* \in \partial J^*(p^*)$. Therefore (u^*, p^*) satisfies the optimality conditions (3.12,3.13) for (PD), which means u^* solves (P) ([Roc70] 31.3). Uniqueness follows from the assumption that H^* is differentiable, which by ([Roc70] 26.3) means that $H(u)$ in the primal functional is strictly convex. □

It will be shown later in Section 3.4.2.4 how to equate modified versions of the PDHG algorithm with convergent alternating direction methods, namely split inexact Uzawa methods from [ZBO09] applied to the split primal ($\mathrm{SP_P}$) and split dual ($\mathrm{SP_D}$) problems. The connection there is very similar to the equivalence from [Tse91] between PFBS applied to (D) and what Tseng in [Tse91] called the alternating minimization algorithm (AMA) applied to ($\mathrm{SP_P}$). Recall from Section 2.3.3.1 that AMA applied to ($\mathrm{SP_P}$) is an alternating direction method that alternately minimizes first the Lagrangian $L_P(u, w, p)$ with respect to u and then the augmented Lagrangian $L_P + \frac{\delta_k}{2}\|Au - w\|_2^2$ with respect to w before

updating the Lagrange multiplier p.

Algorithm: AMA on ($\mathrm{SP_P}$)

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle A^T p^k, u\rangle \tag{3.21a}$$

$$w^{k+1} = \arg\min_{w\in\mathbb{R}^n} J(w) - \langle p^k, w\rangle + \frac{\delta_k}{2}\|Au^{k+1} - w\|_2^2 \tag{3.21b}$$

$$p^{k+1} = p^k + \delta_k(Au^{k+1} - w^{k+1}) \tag{3.21c}$$

To see the equivalence between (3.20) and (3.21), first note that (3.21a) is identical to (3.20a), so it suffices to show that (3.21b) and (3.21c) are together equivalent to (3.20b). Combining (3.21b) and (3.21c) yields

$$p^{k+1} = (p^k + \delta_k Au^{k+1}) - \delta_k \arg\min_w J(w) + \frac{\delta_k}{2}\|w - \frac{(p^k + \delta_k Au^{k+1})}{\delta_k}\|_2^2.$$

By the Moreau decomposition (2.29), this is equivalent to

$$p^{k+1} = \arg\min_p J^*(p) + \frac{1}{2\delta_k}\|p - (p^k + \delta_k Au^{k+1})\|_2^2,$$

which is exactly (3.20b).

In [Tse91], convergence of (u^k, w^k, p^k) satisfying (3.21) to a saddle point of $L_P(u, w, p)$ is directly proved under the assumption that H is strongly convex, an assumption that directly implies the condition on H^* in Theorem 3.4.1.

The other special case of PDHG where $\delta_k = \infty$ can be analyzed in a similar manner. The corresponding algorithm is PFBS applied to (P),

Algorithm: PFBS on (P)

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \langle -Au^k, p\rangle \tag{3.22a}$$

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle u, A^T p^{k+1}\rangle + \frac{1}{2\alpha_k}\|u - u^k\|_2^2, \tag{3.22b}$$

which is analogously equivalent to AMA applied to ($\mathrm{SP_D}$).

Algorithm: AMA on ($\mathrm{SP_D}$)

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \langle -Au^k, p\rangle \tag{3.23a}$$

$$y^{k+1} = \arg\min_{y\in\mathbb{R}^m} H^*(y) - \langle u^k, y\rangle + \frac{\alpha_k}{2}\|y + A^T p^{k+1}\|_2^2 \tag{3.23b}$$

$$u^{k+1} = u^k + \alpha_k(-A^T p^{k+1} - y^{k+1}) \tag{3.23c}$$

The equivalence again follows from the Moreau decomposition (2.29), and the analogous version of Theorem 3.4.1 applies to (3.22).

Note that there are other ways to apply the algorithms described above. For example, when applying PFBS to (P), we could have applied the gradient step to $H(u)$ and the proximal step to $J(Au)$. This would have corresponded to swapping the roles of p and y in AMA applied to ($\mathrm{SP_D}$). There is a corresponding alternate version of AMA on ($\mathrm{SP_P}$). But these alternate versions aren't considered here because they aren't as closely connected to PDHG. In addition, those alternate versions involve more complicated minimization steps in the sense that variables are coupled by either the matrix A or A^T.

3.4.2.2 Reinterpretation of PDHG as Relaxed AMA

The general form of PDHG (3.18) can also be interpreted as alternating direction methods applied to ($\mathrm{SP_P}$) or ($\mathrm{SP_D}$). These interpretations turn out to be relaxed forms of AMA. They can be obtained by modifying the objective functional for the Lagrangian minimization step by adding either $\frac{1}{2\alpha_k}\|u-u^k\|_2^2$ to (3.21a) or $\frac{1}{2\delta_k}\|p-p^k\|_2^2$ to (3.23a).

Algorithm: Relaxed AMA on ($\mathrm{SP_P}$)

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle A^T p^k, u\rangle + \frac{1}{2\alpha_k}\|u-u^k\|_2^2 \tag{3.24a}$$

$$w^{k+1} = \arg\min_{w\in\mathbb{R}^n} J(w) - \langle p^k, w\rangle + \frac{\delta_k}{2}\|Au^{k+1}-w\|_2^2 \tag{3.24b}$$

$$p^{k+1} = p^k + \delta_k(Au^{k+1}-w^{k+1}) \tag{3.24c}$$

Algorithm: Relaxed AMA on ($\mathrm{SP_D}$)

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \langle -Au^k, p\rangle + \frac{1}{2\delta_k}\|p-p^k\|_2^2 \tag{3.25a}$$

$$y^{k+1} = \arg\min_{y\in\mathbb{R}^m} H^*(y) - \langle u^k, y\rangle + \frac{\alpha_k}{2}\|y + A^T p^{k+1}\|_2^2 \tag{3.25b}$$

$$u^{k+1} = u^k + \alpha_k(-A^T p^{k+1} - y^{k+1}) \tag{3.25c}$$

The equivalence of these relaxed AMA algorithms to the general form of PDHG (3.18) follows by a similar argument as in Section 3.4.2.1.

3.4.2.3 Connection to ADMM

Although equating PDHG to the relaxed AMA algorithm doesn't yield any direct convergence results for PDHG, it does show a close connection to ADMM, which does have a well established convergence theory, discussed in Section 2.3.2.3. If, instead of adding proximal terms of the form $\frac{1}{2\alpha_k}\|u-u^k\|_2^2$ and $\frac{1}{2\delta_k}\|p-p^k\|_2^2$ to the first step of AMA applied to ($\mathrm{SP_P}$) and ($\mathrm{SP_D}$), we add the augmented Lagrangian penalties $\frac{\delta_k}{2}\|Au-w^k\|_2^2$ and $\frac{\alpha_k}{2}\|A^T p + y^k\|_2^2$, then we get exactly ADMM applied to ($\mathrm{SP_P}$) and ($\mathrm{SP_D}$) respectively.

Algorithm: ADMM on ($\mathrm{SP_P}$)

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle A^T p^k, u\rangle + \frac{\delta_k}{2}\|Au - w^k\|_2^2 \tag{3.26a}$$

$$w^{k+1} = \arg\min_{w\in\mathbb{R}^n} J(w) - \langle p^k, w\rangle + \frac{\delta_k}{2}\|Au^{k+1} - w\|_2^2 \tag{3.26b}$$

$$p^{k+1} = p^k + \delta_k(Au^{k+1} - w^{k+1}) \tag{3.26c}$$

Algorithm: ADMM on ($\mathrm{SP_D}$)

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \langle -Au^k, p\rangle + \frac{\alpha_k}{2}\|y^k + A^T p\|_2^2 \tag{3.27a}$$

$$y^{k+1} = \arg\min_{y\in\mathbb{R}^m} H^*(y) - \langle u^k, y\rangle + \frac{\alpha_k}{2}\|y + A^T p^{k+1}\|_2^2 \tag{3.27b}$$

$$u^{k+1} = u^k + \alpha_k(-A^T p^{k+1} - y^{k+1}) \tag{3.27c}$$

ADMM applied to ($\mathrm{SP_P}$) can be interpreted as Douglas Rachford splitting [DR56] applied to (D) and ADMM applied to ($\mathrm{SP_D}$) can be interpreted as Douglas Rachford splitting applied to (P) [Gab79, GT89, Eck93, EB92]. Section 2.3.2.2 shows how these connections are made. It is also shown in Section 2.3.2.1 [Ess09, Set09] how to interpret these as the split Bregman algorithm of [GO09]. Assuming that we are most interested in finding a solution u^* to (P), when we apply ADMM to ($\mathrm{SP_P}$), we want to ensure that (u^k, w^k, p^k) converges to a saddle point. Conditions for this are given in Theorem 2.3.3 [EB92].

On the other hand, when applying ADMM to ($\mathrm{SP_D}$), it isn't necessary to insist that (p^k, y^k, u^k) converge to a saddle point if we are only interested in the convergence of $\{u^k\}$ to a solution of (P). Instead we can make use of the convergence theory for the equivalent Douglas Rachford splitting method on (P), see Theorem 2.3.2 [Eck93].

An interesting way to arrive at a version of Douglas Rachford splitting that corresponds exactly to ADMM applied to ($\mathrm{SP_D}$) is to apply ADMM to yet another Lagrangian formulation of (P), namely

$$\max_y \inf_{v,u} L_{P_{DR}}(v, u, y) := J(Av) + H(u) + \langle y, v - u\rangle.$$

This yields an easily implementable way of writing the Douglas Rachford splitting algorithm [Eck93].

Algorithm: Douglas Rachford on (P)

$$v^{k+1} = \arg\min_{v\in\mathbb{R}^m} J(Av) + \frac{1}{2\alpha_k}\|v - u^k + \alpha_k y^k\|_2^2 \tag{3.28a}$$

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \frac{1}{2\alpha_k}\|u - v^{k+1} - \alpha_k y^k\|_2^2 \tag{3.28b}$$

$$y^{k+1} = y^k + \frac{1}{\alpha_k}(v^{k+1} - u^{k+1}) \tag{3.28c}$$

Douglas Rachford splitting applied to (D) can be derived in an analogous manner or by referring to (2.32).

Algorithm: Douglas Rachford on (D)

$$q^{k+1} = \arg\min_{q\in\mathbb{R}^n} H^*(-A^T q) + \frac{1}{2\delta_k}\|q - p^k + \delta_k w^k\|_2^2 \tag{3.29a}$$

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \frac{1}{2\delta_k}\|p - q^{k+1} - \delta_k w^k\|_2^2 \tag{3.29b}$$

$$w^{k+1} = w^k + \frac{1}{\delta_k}(q^{k+1} - p^{k+1}) \tag{3.29c}$$

3.4.2.4 Modifications of PDHG

In this section we show that two slightly modified versions of the PDHG algorithm, denoted PDHGMp and PDHGMu, can be interpreted as a split inexact Uzawa method from [ZBO09] applied to ($\mathrm{SP_P}$) and ($\mathrm{SP_D}$) respectively. In the constant step size case, PDHGMp replaces p^{k+1} in the u^{k+1} step (3.18b) with $2p^{k+1} - p^k$ whereas PDHGMu replaces u^k in the p^{k+1} step (3.18a) with $2u^k - u^{k-1}$.

The variable step size case will also be discussed. The advantage of these modified algorithms is that for appropriate parameter choices they are nearly as efficient as PDHG numerically, and some known convergence results [ZBO09] can be applied. Alternate convergence results for PDHGMu are also proved in [PCB09, CCN09] based on an argument in [Pop80].

The split inexact Uzawa method from [ZBO09] applied to ($\mathrm{SP_D}$) can be thought of as a modification of ADMM. Applying the main idea of the Bregman operator splitting algorithm from [ZBB09], it adds $\frac{1}{2}\langle p-p^k, (\frac{1}{\delta_k}-\alpha_k AA^T)(p-p^k)\rangle$ to the penalty term $\frac{\alpha_k}{2}\|A^Tp+y^k\|_2^2$ in the objective functional for the first minimization step. To ensure $\frac{1}{\delta_k}-\alpha_k AA^T$ is positive definite, choose $0<\delta_k<\frac{1}{\alpha_k\|A\|^2}$. Adding this extra term, like the surrogate functional approach of [DDM04], has the effect of linearizing the penalty term and decoupling the variables previously coupled by the matrix A^T. The updates for y^{k+1} and u^{k+1} remain the same as for ADMM. By combining terms for the p^{k+1} update, the resulting algorithm can be written as

---

Algorithm: Split Inexact Uzawa applied to ($\mathrm{SP_D}$)

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \frac{1}{2\delta_k}\|p-p^k-\delta_k Au^k+\alpha_k\delta_k A(A^Tp^k+y^k)\|_2^2 \tag{3.30a}$$

$$y^{k+1} = \arg\min_{y\in R^m} H^*(y) - \langle u^k, y\rangle + \frac{\alpha_k}{2}\|y+A^Tp^{k+1}\|_2^2 \tag{3.30b}$$

$$u^{k+1} = u^k + \alpha_k(-A^Tp^{k+1}-y^{k+1}). \tag{3.30c}$$

The above algorithm can be shown to converge at least for fixed step sizes α and δ satisfying $0<\delta<\frac{1}{\alpha\|A\|^2}$.

Theorem 3.4.2. *[ZBO09] Let $\alpha_k=\alpha>0$, $\delta_k=\delta>0$ and $0<\delta<\frac{1}{\alpha\|A\|^2}$. Let (p^k,y^k,u^k) satisfy (3.30). Also let p^* be optimal for (D) and $y^*=-A^Tp^*$. Then*

- $\|A^T p^k + y^k\|_2 \to 0$
- $J^*(p^k) \to J^*(p^*)$
- $H^*(y^k) \to H^*(y^*)$

and all convergent subsequences of (p^k, y^k, u^k) *converge to a saddle point of* L_D *(3.17).*

Moreover, the split inexact Uzawa algorithm can be rewritten in a form that is very similar to PDHG. Since the y^{k+1} (3.30b) and u^{k+1} (3.30c) steps are the same as those for AMA on ($\mathrm{SP_D}$) (3.23), then by the same argument they are equivalent to the u^{k+1} update in PDHG (3.18b). From (3.30c), we have that

$$y^k = \frac{u^{k-1}}{\alpha_{k-1}} - \frac{u^k}{\alpha_{k-1}} - A^T p^k. \tag{3.31}$$

Substituting this into (3.30a), we see that (3.30) is equivalent to a modified form of PDHG where u^k is replaced by $\left((1+\frac{\alpha_k}{\alpha_{k-1}})u^k - \frac{\alpha_k}{\alpha_{k-1}}u^{k-1}\right)$ in (3.18a). The resulting form of the algorithm will be denoted PDHGMu.

Algorithm: PDHGMu

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \langle p, -A\left((1+\frac{\alpha_k}{\alpha_{k-1}})u^k - \frac{\alpha_k}{\alpha_{k-1}}u^{k-1}\right)\rangle + \frac{1}{2\delta_k}\|p-p^k\|_2^2 \tag{3.32a}$$

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle A^T p^{k+1}, u\rangle + \frac{1}{2\alpha_k}\|u-u^k\|_2^2, \tag{3.32b}$$

Note that from (3.31) and (3.32b), $y^{k+1} = \partial H(u^{k+1})$, which we could substitute instead of (3.31) into (3.30a) to get an equivalent version of PDHGMu, whose updates only depend on the previous iteration instead of the previous two.

The corresponding split inexact Uzawa method applied to ($\mathrm{SP_P}$) is obtained by adding $\frac{1}{2}\langle u-u^k, (\frac{1}{\alpha_k}-\delta_k A^TA)(u-u^k)\rangle$ to the u^{k+1} step of ADMM applied to ($\mathrm{SP_P}$).

Algorithm: Split Inexact Uzawa applied to ($\mathrm{SP_P}$)

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \frac{1}{2\alpha_k}\|u-u^k+\alpha_k A^Tp^k+\delta_k\alpha_k A^T(Au^k-w^k)\|_2^2 \quad (3.33a)$$

$$w^{k+1} = \arg\min_{w\in\mathbb{R}^n} J(w) - \langle p^k, w\rangle + \frac{\delta_k}{2}\|Au^{k+1}-w\|_2^2 \quad (3.33b)$$

$$p^{k+1} = p^k + \delta_k(Au^{k+1}-w^{k+1}) \quad (3.33c)$$

Again by Theorem 3.4.2, the above algorithm converges for fixed stepsizes α and δ with $0<\alpha<\frac{1}{\delta\|A\|^2}$. Note this requirement is equivalent to requiring $0<\delta<\frac{1}{\alpha\|A\|^2}$.

Since from (3.33c), we have that

$$w^k = \frac{p^{k-1}}{\delta_{k-1}} - \frac{p^k}{\delta_{k-1}} + Au^k, \quad (3.34)$$

a similar argument shows that (3.33) is equivalent to a modified form of PDHG where p^k is replaced by $\left((1+\frac{\delta_k}{\delta_{k-1}})p^k - \frac{\delta_k}{\delta_{k-1}}p^{k-1}\right)$. The resulting form of the algorithm will be denoted PDHGMp.

Algorithm: PDHGMp

$$u^{k+1} = \arg\min_{u\in\mathbb{R}^m} H(u) + \langle A^T\left((1+\frac{\delta_k}{\delta_{k-1}})p^k - \frac{\delta_k}{\delta_{k-1}}p^{k-1}\right), u\rangle + \frac{1}{2\alpha_k}\|u-u^k\|_2^2, \quad (3.35a)$$

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) - \langle p, Au^{k+1}\rangle + \frac{1}{2\delta_k}\|p-p^k\|_2^2 \quad (3.35b)$$

The modifications to u^k and p^k in the split inexact Uzawa methods are reminiscent of the predictor-corrector step in Chen and Teboulle's predictor corrector proximal method (PCPM) [CT94]. Despite some close similarities, however, the algorithms are not equivalent. The modified PDHG algorithms are more implicit than PCPM.

The connections between the algorithms discussed so far are diagrammed in Figure 3.1. For simplicity, constant step sizes are assumed in the diagram.

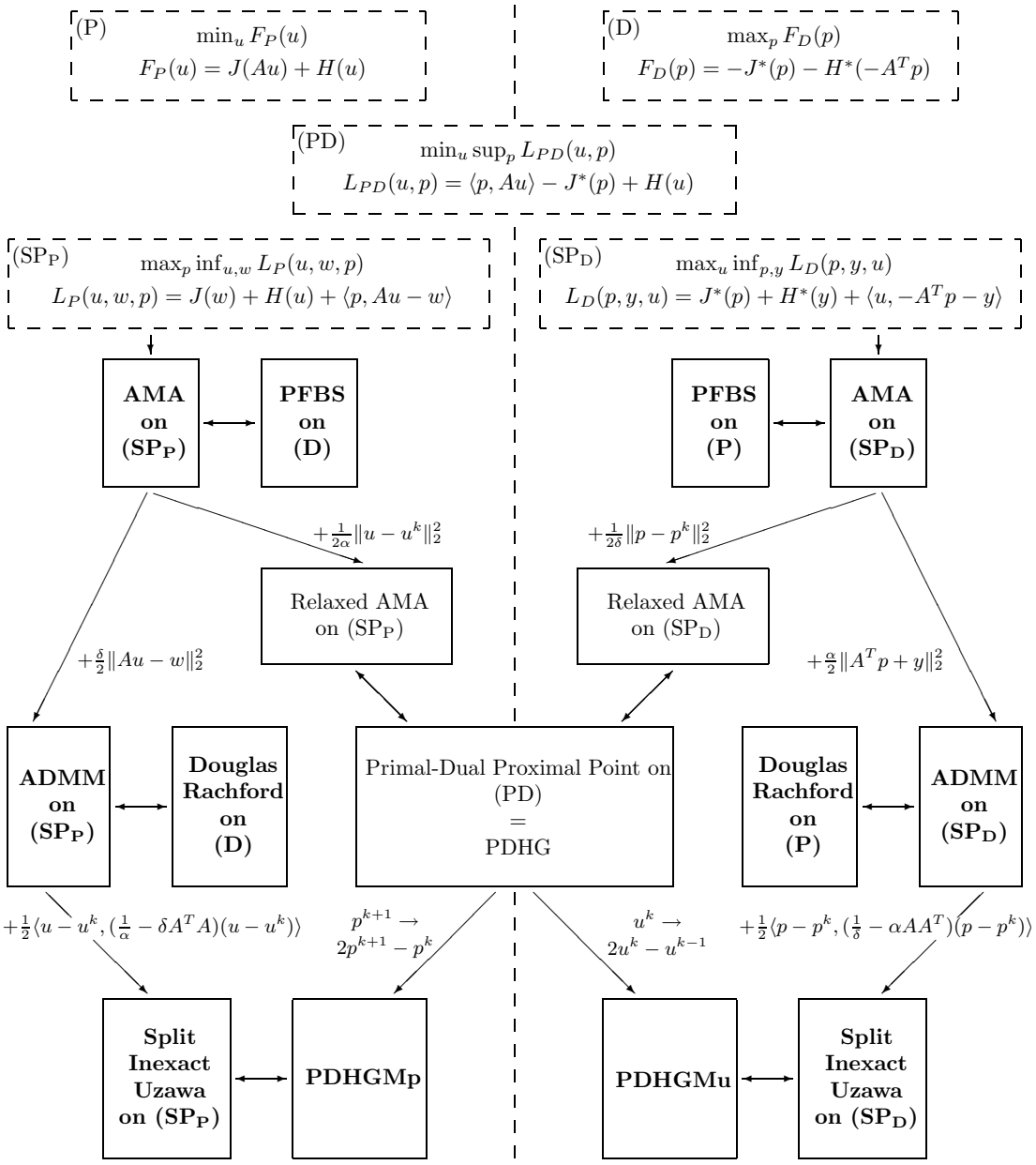
3.5 Interpretation of PDHG as Projected Averaged Gradient Method for TV Denoising

Even though we know of a convergence result (Theorem 3.4.2) for the modified PDHG algorithms PDHGMu (3.32) and PDHGMp, it would be nice to show convergence of the original PDHG method (3.18) because PDHG still has some numerical advantages. Empirically, the stability requirements for the step size parameters are less restrictive for PDHG, so there is more freedom to tune the parameters to improve the rate of convergence. In this section, we restrict attention to PDHG applied to TV denoising and prove a convergence result assuming certain conditions on the parameters.

3.5.1 Projected Gradient Special Case

In the case of TV denoising, problem (P) becomes

$$\min_{u\in\mathbb{R}^m} \|u\|_{TV} + \frac{\lambda}{2}\|u-f\|_2^2, \tag{3.36}$$



Legend:
(P): Primal
(D): Dual
(PD): Primal-Dual
(SP_P): Split Primal
(SP_D): Split Dual

AMA: Alternating Minimization Algorithm (4.2.1)
PFBS: Proximal Forward Backward Splitting (4.2.1)
ADMM: Alternating Direction Method of Multipliers (4.2.2)
PDHG: Primal Dual Hybrid Gradient (4.2)
PDHGM: Modified PDHG (4.2.3)
Bold: Well Understood Convergence Properties

Figure 3.1: PDHG-Related Algorithm Framework

with $J = \|\cdot\|_E$, $A = D$ and $H(u) = \frac{\lambda}{2}\|u - f\|_2^2$, in which case PFBS on (D) simplifies to

$$p^{k+1} = \arg\min_{p\in\mathbb{R}^n} J^*(p) + \frac{1}{2\delta_k}\|p - (p^k + \delta_k D\nabla H^*(-D^T p^k))\|_2^2.$$

Since J^* is the indicator function for the unit ball, denoted X (3.3), in the dual norm $\|\cdot\|_{E^*}$, this is exactly an orthogonal projection onto the convex set X (3.10). Letting $\tau_k = \frac{\delta_k}{\lambda}$ and using also that

$$H^*(-D^T p) = \frac{1}{2\lambda}\|\lambda f - D^T p\|_2^2 - \frac{\lambda}{2}\|f\|_2^2,$$

the algorithm simplifies to

Algorithm: Gradient Projection for TV Denoising

$$p^{k+1} = \Pi_X\left(p^k - \tau_k D(D^T p^k - \lambda f)\right). \tag{3.37}$$

Many variations of gradient projection applied to TV denoising are discussed in [ZWC08]. As already noted in [ZC08], algorithm PDGH applied to TV denoising reduces to projected gradient descent when $\theta_k = 1$. Equivalence to (3.9) in the $\theta_k = 1$ case can be seen by plugging $u^k = (f - \frac{1}{\lambda}D^T p^k)$ into the update for p^{k+1}. This can be interpreted as projected gradient descent applied to

$$\min_{p\in X} G(p) := \frac{1}{2}\|D^T p - \lambda f\|_2^2, \tag{3.38}$$

an equivalent form of the dual problem.

Theorem 3.5.1. *Fix $p^0 \in \mathbb{R}^n$. Let p^k be defined by (3.37) with $0 < \inf \tau_k \leq \sup \tau_k < \frac{1}{4}$, and define $u^{k+1} = f - \frac{D^T p^k}{\lambda}$. Then $\{p^k\}$ converges to a solution of (3.38), and $\{u^k\}$ converges to a solution of (3.36).*

Proof. Since ∇G is Lipschitz continuous with Lipschitz constant $\|DD^T\|$ and $u^{k+1} = \nabla H^*(-D^T p^k) = f - \frac{D^T p^k}{\lambda}$, then by Theorem 3.4.1 the result follows if $0 < \inf \tau_k \leq \sup \tau_k < \frac{2}{\|DD^T\|}$. The bound $\|DD^T\| \leq 8$ follows from the Gersgorin circle theorem. □

3.5.1.1 AMA Equivalence and Soft Thresholding Interpretation

By the general equivalence between PFBS and AMA discussed in Section 2.3.3.1, the gradient projection algorithm (3.37) is equivalent to

---

Algorithm: AMA for TV Denoising

$$u^{k+1} = f - \frac{D^T p^k}{\lambda} \tag{3.39a}$$

$$w^{k+1} = \tilde{S}_{\frac{1}{\delta_k}}(Du^{k+1} + \frac{1}{\delta_k} p^k) \tag{3.39b}$$

$$p^{k+1} = p^k + \delta_k(Du^{k+1} - w^{k+1}), \tag{3.39c}$$

---

where $\tilde{S}$ (2.54) denotes the soft thresholding operator for $\|\cdot\|_E$. It can be interpreted in terms of an orthogonal projection onto the set X.

$$\tilde{S}_\alpha(f) = \arg\min_z \|z\|_E + \frac{1}{2\alpha}\|z - f\|_2^2 = f - \Pi_{\alpha X}(f). \tag{3.40}$$

Similar formulas to (3.40) arise when J equals norms other than $\|\cdot\|_E$, the modification being that X is replaced by the unit ball in the respective dual norms. In fact, it's not always necessary to assume that J is a norm to obtain similar projection interpretations. It's enough that J be a convex 1-homogeneous function, as Chambolle points out in [Cha04] when deriving a projection formula for the solution of the TV denoising problem. By letting $z = D^T p$, the dual

problem (3.38) is solved by the projection

$$z = \Pi_{\{z:z=D^T p, \|p\|_{E^*} \leq 1\}}(\lambda f),$$

and the solution to the TV denoising problem is given by

$$u^* = f - \frac{1}{\lambda}\Pi_{\{z:z=D^T p, \|p\|_{E^*} \leq 1\}}(\lambda f).$$

However, the projection is nontrivial to compute.

3.5.2 Projected Averaged Gradient

In the $\theta \neq 1$ case, still for TV denoising, the projected gradient descent interpretation of PDHG extends to an interpretation as a projected averaged gradient descent algorithm. Consider for simplicity parameters τ and θ that are independent of k. Then plugging u^{k+1} into the update for p yields

$$p^{k+1} = \Pi_X \left(p^k - \tau d_\theta^k\right) \tag{3.41}$$

where

$$d_\theta^k = \theta \sum_{i=1}^{k} (1-\theta)^{k-i} \nabla G(p^i) + (1-\theta)^k \nabla G(p^0)$$

is a convex combination of gradients of G at the previous iterates p^i. Note that d_θ^k is not necessarily a descent direction.

This kind of averaging of previous iterates suggests a connection to Nesterov's method [Nes07]. Several recent papers study variants of his method and their applications. Weiss, Aubert and Blanc-Féraud in [WAB07] apply a variant of Nesterov's method [Nes05] to smoothed TV functionals. Beck and Teboulle in [BT09] and Becker, Bobin and Candes in [BBC] also study variants of Nesterov's method that apply to l_1 and TV minimization problems. Tseng gives a unified treatment of accelerated proximal gradient methods like Nesterov's in [Tse08].

However, despite some tantalizing similarities to PDHG, it appears that none is equivalent.

In the following section, the connection to a projected average gradient method on the dual is made for the more general case when the parameters are allowed to depend on k. Convergence results are presented for some special cases.

3.5.2.1 Convergence

For a minimizer $\overline{p}$, the optimality condition for the dual problem (3.38) is

$$\overline{p} = \Pi_X(\overline{p} - \tau \nabla G(\overline{p})), \quad \forall \tau \geq 0, \tag{3.42}$$

or equivalently

$$\langle \nabla G(\overline{p}), p - \overline{p} \rangle \geq 0, \quad \forall p \in X.$$

In the following, we denote $\overline{G} = \min_{p \in X} G(p)$ and let X^* denote the set of minimizers. As mentioned above, the PDHG algorithm (3.9) for TV denoising is related to a projected gradient method on the dual variable p. When τ and θ are allowed to depend on k, the algorithm can be written as

$$p^{k+1} = \Pi_X \left(p^k - \tau_k d^k \right) \tag{3.43}$$

where

$$d^k = \sum_{i=0}^{k} s_k^i \nabla G(p^i), \quad s_k^i = \theta_{i-1} \prod_{j=i}^{k-1} (1 - \theta_j).$$

Note that

$$\sum_{i=0}^{k} s_k^i = 1, \quad s_k^i = (1 - \theta_{k-1}) s_{k-1}^i \qquad \forall k \geq 0, i \leq k, \qquad \text{and} \tag{3.44}$$

$$d^k = (1 - \theta_{k-1}) d^{k-1} + \theta_{k-1} \nabla G(p^k). \tag{3.45}$$

As above, the direction d^k is a linear (convex) combination of gradients of all previous iterates. We will show d^k is an ϵ-gradient at p^k. This means d^k is an

element of the ϵ-differential (ϵ-subdifferential for nonsmooth functionals), $\partial_\epsilon G(p)$, of G at p^k defined by

$$G(q) \geq G(p^k) + \langle d^k, q - p^k \rangle - \epsilon, \forall q \in X$$

When $\epsilon = 0$ this is the definition of d^k being a sub-gradient (in this case, the gradient) of G at p^k.

For p and q, the Bregman distance based on G between p and q is defined as

$$D(p, q) = G(p) - G(q) - \langle \nabla G(q), p - q \rangle \quad \forall p, q \in X \tag{3.46}$$

From (3.38), the Bregman distance (3.46) reduces to

$$D(p, q) = \frac{1}{2}\|D^T(p - q)\|_2^2 \leq \frac{L}{2}\|p - q\|^2,$$

where L is the Lipschitz constant of ∇G.

Lemma 3.5.1. *For any $q \in X$, we have*

$$G(q) - G(p^k) - \langle d^k, q - p^k \rangle = \sum_{i=0}^{k} s_k^i (D(q, p^i) - D(p^k, p^i)).$$

Proof. For any $q \in X$,

$$\begin{aligned}
G(q) - G(p^k) - \langle d^k, q - p^k \rangle &= G(q) - G(p^k) - \langle \sum_{i=0}^{k} s_k^i \nabla G(p^i), q - p^k \rangle \\
&= \sum_{i=0}^{k} s_k^i G(q) - \sum_{i=0}^{k} s_k^i G(p^i) - \sum_{i=0}^{k} s_k^i \langle \nabla G(p^i), q - p^i \rangle \\
&\quad + \sum_{i=0}^{k} s_k^i (G(p^i) - G(p^k) - \langle \nabla G(p^i), p^i - p^k \rangle) \\
&= \sum_{i=0}^{k} s_k^i (D(q, p^i) - D(p^k, p^i))
\end{aligned}$$

□

Lemma 3.5.2. *The direction d^k is a ϵ_k-gradient of p^k where $\epsilon_k = \sum_{i=0}^{k} s_k^i D(p^k, p^i)$.*

Proof. By Lemma 3.5.1,

$$G(q) - G(p^k) - \langle d^k, q - p^k \rangle \geq -\sum_{i=0}^{k} s_k^i D(p^k, p^i) \quad \forall q \in X.$$

By the definition of ϵ-gradient, we obtain that d^k is a ϵ_k-gradient of G at p^k, where

$$\epsilon_k = \sum_{i=0}^{k} s_k^i D(p^k, p^i).$$

□

Lemma 3.5.3. *If $\theta_k \to 1$, then $\epsilon_k \to 0$.*

Proof. Let $h_k = G(p^k) - G(p^{k-1}) - \langle d^{k-1}, p^k - p^{k-1} \rangle$, then using the Lipschitz continuity of ∇G and the boundedness of d^k, we obtain

$$|h_k| = |D(p^k, p^{k-1}) + \langle (\nabla G(p^{k-1}) - d^{k-1}, p^k - p^{k-1} | \rangle| \leq \frac{L}{2} \|p^k - p^{k-1}\|_2^2 + C_1 \|p^k - p^{k-1}\|_2,$$

where L is the Lipschitz constant of ∇G, and C_1 is some positive constant. Since $\epsilon_k = \sum_{i=0}^{k} s_k^i D(p^k, p^i)$, and $\sum_{i=0} s_k^i = 1$, then ϵ_k is bounded for any k.

Meanwhile, by replacing q with p^k and p^k by p^{k-1} in Lemma 3.5.1, we obtain $h_k = \sum_{i=0}^{k-1} s_{k-1}^i (D(p^k, p^i) - D(p^{k-1}, p^i))$. From

$$s_k^i = (1 - \theta_{k-1}) s_{k-1}^i, \ \forall \ 1 \leq i \leq k-1,$$

we get

$$\begin{aligned} \epsilon_k &= (1 - \theta_{k-1}) \sum_{i=0}^{k-1} s_{k-1}^i D(p^k, p^i) \\ &= (1 - \theta_{k-1}) \epsilon_{k-1} + (1 - \theta_{k-1}) \sum_{i=0}^{k-1} s_{k-1}^i (D(p^k, p^i) - D(p^{k-1}, p^i)) \\ &= (1 - \theta_{k-1})(\epsilon_{k-1} + h_k). \end{aligned}$$

By the boundness of h_k and ϵ_k, we get immediately that if $\theta_{k-1} \to 1$, then $\epsilon_k \to 0$. □

Since $\epsilon_k \to 0$, the convergence of p^k follows directly from classical [SKR85, LPS03] ϵ-gradient methods. Possible choices of the step size τ_k are given in the following theorem:

Theorem 3.5.2. *[SKR85, LPS03][Convergence to the optimal set using divergent series τ_k] Let $\theta_k \to 1$ and let τ_k satisfy $\tau_k > 0$, $\lim_{k\to\infty} \tau_k = 0$ and $\sum_{k=1}^{\infty} \tau_k = \infty$. Then the sequence p^k generated by (3.43) satisfies $G(p^k) \to \overline{G}$ and dist$\{p^k, X^*\} \to 0$.*

Since we require $\theta_k \to 1$, the algorithm is equivalent to projected gradient descent in the limit. However, it is well known that a divergent step size for τ_k is slow and we can expect a better convergence rate without letting τ_k go to 0. In the following, we prove a different convergence result that doesn't require $\tau_k \to 0$ but still requires $\theta_k \to 1$.

Lemma 3.5.4. *For p^k defined by (3.43), we have $\langle d^k, p^{k+1} - p^k \rangle \leq -\frac{1}{\tau_k} \|p^{k+1} - p^k\|_2^2$.*

Proof. Since p^{k+1} is the projection of $p^k - \tau_k d^k$ onto X, it follows that

$$\langle p^k - \tau_k d^k - p^{k+1}, p - p^{k+1} \rangle \leq 0, \quad \forall p \in X.$$

Replacing p with p^k, we thus get

$$\langle d^k, p^{k+1} - p^k \rangle \leq -\frac{1}{\tau_k} \|p^{k+1} - p^k\|_2^2.$$

□

Lemma 3.5.5. *Let p^k be generated by the method (3.43), then*

$$\begin{aligned} &G(p^{k+1}) - G(p^k) - \frac{\beta_k^2}{\alpha_k}\|p^k - p^{k-1}\|_2^2 \\ &\leq -\frac{(\alpha_k+\beta_k)^2}{\alpha_k}\|p^k - (\frac{\alpha_k}{\alpha_k+\beta_k}p^{k+1} + \frac{\beta_k}{\alpha_k+\beta_k}p^{k-1})\|_2^2 \end{aligned}$$

where

$$\alpha_k = \frac{1}{\tau_k\theta_{k-1}} - \frac{L}{2}, \quad \beta_k = \frac{1-\theta_{k-1}}{2\theta_{k-1}\tau_{k-1}} \tag{3.47}$$

Proof. By using the Taylor expansion and the Lipschiz continuity of ∇G (or directly from the fact that G is quadratic function), we have

$$G(p^{k+1}) - G(p^k) \leq \langle \nabla G(p^k), p^{k+1} - p^k\rangle + \frac{L}{2}\|p^{k+1} - p^k\|_2^2,$$

Since $\nabla G(p^k) = \frac{1}{\theta_{k-1}}(d^k - (1-\theta_{k-1})d^{k-1})$, we have

$$\begin{aligned} G(p^{k+1}) - G(p^k) &\leq \frac{1}{\theta_{k-1}}\langle d^k, p^{k+1} - p^k\rangle - \frac{1-\theta_{k-1}}{\theta_{k-1}}\langle d^{k-1}, p^{k+1} - p^k\rangle \\ &\quad + \frac{L}{2}\|p^{k+1} - p^k\|_2^2, \\ &= (\frac{L}{2} - \frac{1}{\tau_k\theta_{k-1}})\|p^{k+1} - p^k\|_2^2 - \frac{1-\theta_{k-1}}{\theta_{k-1}}\langle d^{k-1}, p^{k+1} - p^k\rangle. \end{aligned}$$

On the other hand, since p^k is the projection of $p^{k-1} - \tau_{k-1}d^{k-1}$, we get

$$\langle p^{k-1} - \tau_{k-1}d^{k-1} - p^k, p - p^k\rangle \leq 0, \quad \forall p \in X.$$

Replacing p with p^{k+1}, we thus get

$$\langle d^{k-1}, p^{k+1} - p^k\rangle \geq \frac{1}{\tau_{k-1}}\langle p^{k-1} - p^k, p^{k+1} - p^k\rangle.$$

This yields

$$\begin{aligned} G(p^{k+1}) - G(p^k) &\leq -\alpha_k\|p^{k+1} - p^k\|^2 - 2\beta_k\langle p^{k-1} - p^k, p^{k+1} - p^k\rangle \\ &= -\frac{(\alpha_k+\beta_k)^2}{\alpha_k}\|p^k - (\frac{\alpha_k}{\alpha_k+\beta_k}p^{k+1} + \frac{\beta_k}{\alpha_k+\beta_k}p^{k-1})\|^2 \\ &\quad + \frac{\beta_k^2}{\alpha_k}\|p^k - p^{k-1}\|^2. \end{aligned}$$

where α_k and β_k are defined as (3.47).

□

Theorem 3.5.3. *If α_k and β_k defined as (3.47) such that $\alpha_k > 0, \beta_k \geq 0$ and*

$$\sum_{k=0}^{\infty} \frac{(\alpha_k + \beta_k)^2}{\alpha_k} = \infty, \quad \sum_{k=0}^{\infty} \frac{\beta_k^2}{\alpha_k} < \infty, \quad \lim_{k\to\infty} \frac{\beta_k}{\alpha_k} = 0. \tag{3.48}$$

then every limit point pair (p^∞, d^∞) of a subsequence of (p^k, d^k) is such that p^∞ is a minimizer of (3.38) and $d^\infty = \nabla G(p^\infty)$.

Proof. The proof is adapted from [Ber99](Proposition 2.3.1,2.3.2) and Lemma 3.5.5. Since p^k and d^k are bounded, the subsequence (p^k, d^k) has a convergent subsequence. Let (p^∞, d^∞) be a limit point of the pair (p^k, d^k), and let (p^{k_m}, d^{k_m}) be a subsequence that converges to (p^∞, d^∞). For $k_m > n_0$, lemma 3.5.5 implies that

$$\begin{aligned} G(p^{k_m}) - G(p^{n_0}) \leq &- \sum_{k=n_0}^{k_m} \frac{(\alpha_k + \beta_k)^2}{\alpha_k} \|p^k - (\frac{\alpha_k}{\alpha_k + \beta_k} p^{k+1} + \frac{\beta_k}{\alpha_k + \beta_k} p^{k-1})\|_2^2 \\ &+ \sum_{k=n_0}^{k_m} \frac{\beta_k^2}{\alpha_k} \|p^{k-1} - p^k\|_2^2. \end{aligned}$$

By the boundness of the constraint set X, the conditions (3.48) for α_k and β_k and the fact that $G(p)$ is bounded from below, we conclude that

$$\|p^k - (\frac{\alpha_k}{\alpha_k + \beta_k} p^{k+1} + \frac{\beta_k}{\alpha_k + \beta_k} p^{k-1})\|_2 \to 0.$$

Given $\epsilon > 0$, we can choose m large enough such that $\|p^{k_m} - p^\infty\|_2 \leq \frac{\epsilon}{3}$, $\|p^k - (\frac{\alpha_k}{\alpha_k+\beta_k} p^{k+1} + \frac{\beta_k}{\alpha_k+\beta_k} p^{k-1})\|_2 \leq \frac{\epsilon}{3}$ for all $k \geq k_m$, and $\frac{\beta_{k_m}}{\alpha_{k_m}+\beta_{k_m}} \|(p^{k_m-1} - p^\infty)\|_2 \leq \frac{\epsilon}{3}$. This third requirement is possible because $\lim_{k\to\infty} \frac{\beta_k}{\alpha_k} = 0$. Then

$$\|(p^{k_m} - p^\infty) - \frac{\alpha_{k_m}}{\alpha_{k_m} + \beta_{k_m}} (p^{k_m+1} - p^\infty) - \frac{\beta_{k_m}}{\alpha_{k_m} + \beta_{k_m}} (p^{k_m-1} - p^\infty)\|_2 \leq \frac{\epsilon}{3}$$

implies

$$\|\frac{\alpha_{k_m}}{\alpha_{k_m}+\beta_{k_m}}(p^{k_m+1}-p^\infty)+\frac{\beta_{k_m}}{\alpha_{k_m}+\beta_{k_m}}(p^{k_m-1}-p^\infty)\|_2 \leq \frac{2}{3}\epsilon.$$

Since $\frac{\beta_{k_m}}{\alpha_{k_m}+\beta_{k_m}}\|(p^{k_m-1}-p^\infty)\|_2 \leq \frac{\epsilon}{3}$, we have

$$\|p^{k_m+1}-p^\infty\|_2 \leq \frac{\alpha_{k_m}+\beta_{k_m}}{\alpha_{k_m}}\epsilon.$$

Note that k_m+1 is not necessarily an index for the subsequence $\{p^{k_m}\}$. Since $\lim_k \frac{\alpha_k+\beta_k}{\alpha_k} = 1$, then we have $\|p^{k_m+1}-p^\infty\|_2 \to 0$ when $m \to \infty$. According (3.43), the limit point p^∞, d^∞ is therefore such that

$$p^\infty = \Pi_X(p^\infty - \tau d^\infty) \tag{3.49}$$

for $\tau > 0$.

It remains to show that the corresponding subsequence $d^{k_m} = (1-\theta_{k_m-1})d^{k_m-1}+\theta_{k_m-1}\nabla G(p^{k_m})$ converges to $\nabla G(p^\infty)$. By the same technique, and the fact that $\theta_k \to 1$, we can get $\|\nabla G(p^{k_m})-d^\infty\| \leq \epsilon$. Thus $\nabla G(p^{k_m}) \to d^\infty$. On the other hand, $\nabla G(p^{k_m}) \to \nabla G(p^\infty)$. Thus $d^\infty = \nabla G(p^\infty)$. Combining with (3.49) and the optimal condition (3.42), we conclude that p^∞ is a minimizer. □

In summary, the overall conditions on θ_k and τ_k are:

- $\theta_k \to 1, \tau_k > 0$,
- $0 < \tau_k\theta_k < \frac{2}{L}$,
- $\sum_{k=0}^\infty \frac{(\alpha_k+\beta_k)^2}{\alpha_k} = \infty$,
- $\lim_{k\to\infty}\frac{\beta_k}{\alpha_k} = 0$,
- $\sum_{k=0}^\infty \frac{\beta_k^2}{\alpha_k} < \infty$,

where

$$\alpha_k = \frac{1}{\tau_k \theta_{k-1}} - \frac{L}{2}, \quad \beta_k = \frac{1-\theta_{k-1}}{2\theta_{k-1}\tau_{k-1}}. \tag{3.50}$$

Finally, we have $\theta_k \to 1$, and for τ_k the classical condition for the projected gradient descent algorithm, $(0 < \tau_k < \frac{2}{L})$, and divergent stepsize, $(\lim_k \tau_k \to 0, \sum_k \tau_k \to \infty)$, are special cases of the above conditions. Note that even though the convergence with $0 < \theta_k \leq c < 1$ and even $\theta_k \to 0$ is numerically demonstrated in [ZC08], a theoretical proof is still an open problem.

3.6 Applications

In this section we discuss the general types of functionals to which PDHG and it's variants can be applied. We show how seemingly more complicated models can often still be written as the primal problem (P) and give several examples.

3.6.1 General Application to Convex Programs with Separable Structure

Analogous to the approach discussed in 2.1 about rewriting convex programs in the form of (P0), similar operator splitting methods can be used to cast a large class of problems in the form of (P). We will consider functionals that are sums of convex functions composed with linear operators and subject to any number of convex constraints. In particular, consider the problem of minimizing with respect to u

$$F(u) = \sum_{i=1}^{N} \phi_i(B_i A_i u + b_i) + H(u), \tag{3.51}$$

where ϕ_i and H are closed proper convex functions. By defining $A = \begin{bmatrix} A_1 \\ \vdots \\ A_N \end{bmatrix}$ and $J(Au) = \sum_{i=1}^{N} J_i(A_i u)$ with $J_i(z_i) = \phi_i(B_i z_i + b_i)$, we can rewrite

$$\begin{aligned} F(u) &= \sum_{i=1}^{N} J_i(A_i u) + H(u) \\ &= J(Au) + H(u), \end{aligned} \tag{3.52}$$

which is of the form (P). The reason PDHG and it's variants are still effective for such problems is that the J_i terms naturally decouple when the algorithms are applied. Letting $p = \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix}$, it follows that

$$J^*(p) = \sum_{i=1}^{N} J_i^*(p_i).$$

Application of PDHG to (3.52) yields

$$u^{k+1} = \arg\min_u H(u) + \frac{1}{2\alpha_k} \left\| u - \left(u^k - \alpha_k \sum_{i=1}^{N} A_i^T p_i^k \right) \right\|_2^2 \tag{3.53a}$$

$$p_i^{k+1} = \arg\min_{p_i} J_i^*(p_i) + \frac{1}{2\delta_k} \left\| p_i - \left(p_i^k + \delta_k A_i u^{k+1} \right) \right\|_2^2 \qquad i = 1, ..., N. \tag{3.53b}$$

The application of PDHGMp to (3.52) with fixed time steps α and δ additionally replaces p_i^k in the u^{k+1} update with $2p_i^k - p_i^{k-1}$. Recall that for PDHGMp to converge, the parameters must satisfy $\alpha > 0$, $\delta > 0$ and $\alpha\delta < \frac{1}{\|A\|^2}$.

The algorithm is most efficient when the individual minimization steps can be explicitly solved. Some important examples of the types of functionals that make this possible are when $F(u)$ is composed of l_2, l_∞ and l_1 norms, the l_2 norm

squared, l_1-like terms such as TV, l_∞-like terms such as max, indicator functions

$$g_S(u) = \begin{cases} 0 & u \in S \\ \infty & \text{otherwise} \end{cases}$$

for convex sets S that are easy to orthogonally project onto, and any of these functions composed with linear operators. Additionally, the matrices B_i in (3.51) should be chosen so that functionals of the form

$$\phi_i(B_i z + b) + \frac{1}{2}\|z - f\|_2^2$$

are easy to minimize with respect to z. Often this means choosing B_i to be diagonal, which can still be useful for scaling purposes. If $\phi_i = \frac{1}{2}\|\cdot\|_2^2$ for example, then the algorithm remains efficient for B_i where $I + B^T B$ is easily invertible. The l_2 norm squared terms are easy to deal with because they lead to quadratic functions which can be explicitly minimized. The l_2, l_1 and l_∞ norms are easy to deal with because their Legendre transforms are the indicator functions for the unit balls in the l_2, l_∞ and l_1 norms respectively. It's straightforward to orthogonally project onto these convex sets. In the cases of the l_2 and l_∞ unit balls

$$\Pi_{\{z:\|z\|_2 \leq 1\}}(f) = \frac{f}{\max(\|f\|_2, 1)} \tag{3.54}$$

and

$$\Pi_{\{z:\|z\|_\infty \leq 1\}}(f) = \frac{f}{\max(|f|, 1)}, \tag{3.55}$$

where the division in $\Pi_{\{z:\|z\|_\infty \leq 1\}}$ is understood in a componentwise sense. Although there isn't a formula for the orthogonal projection onto the l_1 unit ball, $\Pi_{\{z:\|z\|_1 \leq 1\}}(f)$ can be computed in $O(m \log(m))$ complexity for $f \in \mathbb{R}^m$. Total variation terms, as seen in previous examples, lead to the orthogonal projection onto $\{z : \|z\|_{E^*} \leq 1\}$ defined by (3.10). The Legendre transform of the max function is the indicator function for the positive face of the l_1 unit ball. The

projection onto this convex set is similar to the projection onto the l_1 unit ball and can also be computed with complexity $O(m \log(m))$. This projection appears in Section 3.6.4 and plays an important role in the convex registration and nonlocal inpainting examples in Chapters 4 and 5.

The extension of PDHG to constrained minimization problems is discussed in [ZC08] and applied for example to TV denoising with a constraint of the form $\|u - f\|^2 \leq \sigma^2$ with σ^2 an estimate of the variance of the Gaussian noise. Suppose the constraint on u is of the form $\|Ku - f\|_2 \leq \epsilon$ for some matrix K and $\epsilon > 0$. Let $S = \{u : \|Ku - f\|_2 \leq \epsilon\}$. When $H(u) = g_S(u)$, applying PDHG or the modified versions results in a primal step that can be interpreted as an orthogonal projection onto S. For this to be practical, Π_S must be straightforward to compute. In general for this constraint,

$$\Pi_S(z) = (I - K^\dagger K)z + K^\dagger \begin{cases} Kz & \text{if } \|Kz - f\|_2 \leq \epsilon \\ f + r\left(\frac{Kz - KK^\dagger f}{\|Kz - KK^\dagger f\|_2}\right) & \text{otherwise} \end{cases},$$

where

$$r = \sqrt{\epsilon^2 - \|(I - KK^\dagger)f\|_2^2}$$

and $K^\dagger$ denotes the pseudoinverse of K. Note that $(I - K^\dagger K)$ represents the orthogonal projection onto $\ker(K)$. A special case where this projection is easily computed is when $K = R\Phi$ where R is a row selector and Φ is orthogonal. Then $KK^T = I$ and $K^\dagger = K^T$. In this case, the projection onto S simplifies to

$$\Pi_S(z) = (I - K^T K)z + K^T \begin{cases} Kz & \text{if } \|Kz - f\|_2 \leq \epsilon \\ f + \epsilon\left(\frac{Kz - f}{\|Kz - f\|_2}\right) & \text{otherwise} \end{cases}.$$

Without this kind of simplification, it's often better to define $T = \{z : \|z - f\|_2 \leq \epsilon\}$ and replace $g_S(u)$ with $g_T(Ku)$. By letting one of the J_i terms equal $g_T(z)$

it's only necessary to project onto T, which is significantly easier than projecting onto S. This projection Π_T is defined by

$$\Pi_T(z) = f + \frac{z - f}{\max\left(\frac{\|z-f\|_2}{\epsilon}, 1\right)}. \tag{3.56}$$

3.6.2 Constrained and Unconstrained TV deblurring

In the notation of problem (P), the unconstrained TV deblurring problem (3.1) corresponds to $J = \|\cdot\|_E$, $A = D$ and $H(u) = \frac{\lambda}{2}\|Ku - f\|_2^2$. PDHG applied to (3.1) gives the following algorithm.

Algorithm: PDHG for Unconstrained TV Deblurring

$$u^{k+1} = (\frac{1}{\alpha_k} + \lambda K^T K)^{-1}\left(\lambda K^T f - D^T p^k + \frac{u^k}{\alpha_k}\right)$$
$$p^{k+1} = \Pi_X\left(p^k + \delta_k D u^{k+1}\right),$$

In the case when $\alpha_k + \lambda K^T K$ is not easy to invert, we can instead let

$$H(u) = 0 \qquad \text{and} \qquad J(Au) = J_1(Du) + J_2(Ku),$$

where $A = \begin{bmatrix} D \\ K \end{bmatrix}$, $J_1(w) = \|w\|_E$ and $J_2(z) = \frac{\lambda}{2}\|z - f\|_2^2$. Letting $p = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$, it follows that $J^*(p) = J_1^*(p_1) + J_2^*(p_2)$. Applying PDHG and using the fact that $J_2^*(p_2) = \frac{\|p_2\|_2^2}{2\lambda} + \langle p_2, f\rangle$, we obtain

Algorithm: PDHG for Unconstrained TV Deblurring

$$u^{k+1} = u^k - \alpha_k(D^T p_1^k + K^T p_2^k)$$
$$p_1^{k+1} = \Pi_X\left(p_1^k + \delta_k D u^{k+1}\right)$$
$$p_2^{k+1} = \left(\frac{\delta_k \lambda}{\delta_k + \lambda}\right)\left(\frac{p_2^k}{\delta_k} + K u^{k+1} - f\right).$$

A constrained version of this problem,

$$\min_{\|Ku-f\|_2 \le \epsilon} \|u\|_{TV}, \tag{3.59}$$

can be rewritten as

$$\min_u \|Du\|_E + g_T(Ku).$$

Again let

$$H(u) = 0 \qquad \text{and} \qquad J(Au) = J_1(Du) + J_2(Ku),$$

where $A = \begin{bmatrix} D \\ K \end{bmatrix}$, $J_1(w) = \|w\|_E$ but now $J_2(z) = g_T(z)$. Applying PDHG (3.18) with the u^{k+1} step written first, we obtain

Algorithm: PDHG for Constrained TV Deblurring

$$u^{k+1} = u^k - \alpha_k(D^T p_1^k + K^T p_2^k) \tag{3.60a}$$
$$p_1^{k+1} = \Pi_X\left(p_1^k + \delta_k D u^{k+1}\right) \tag{3.60b}$$
$$p_2^{k+1} = p_2^k + \delta_k K u^{k+1} - \delta_k \Pi_T\left(\frac{p_2^k}{\delta_k} + K u^{k+1}\right). \tag{3.60c}$$

In the constant step size case, to get the PDHGMp version of this algorithm, we would replace $D^T p_1^k + K^T p_2^k$ with $D^T(2p_1^k - p_1^{k-1}) + K^T(2p_2^k - p_2^{k-1})$. Note that λ in the unconstrained problem is related to ϵ by $\lambda = \frac{\|p_2^*\|}{\epsilon}$.

3.6.3 Constrained l_1-Minimization

Compressive sensing problems [CRT05] that seek to find a sparse solution satisfying some data constraints sometimes use the type of constraint described in the previous section. A simple example of such a problem is

$$\min_{z\in\mathbb{R}^m} \|\Psi z\|_1 \qquad \text{such that} \qquad \|R\Gamma z - f\|_2 \leq \epsilon, \tag{3.61}$$

where Ψz is what we expect to be sparse, R is a row selector and Γ is orthogonal. $R\Gamma$ can be thought of as a measurement matrix that represents a selection of some coefficients in an orthonormal basis. We could apply the same strategy used for constrained TV deblurring, but we will instead let Ψ be orthogonal and focus on a simpler example in order to compare two different applications of PDHGMu, one that stays on the constraint set and one that doesn't. Since Ψ is orthogonal, problem (3.61) is equivalent to

$$\min_{u\in\mathbb{R}^m} \|u\|_1 \qquad \text{such that} \qquad \|Ku - f\|_2 \leq \epsilon, \tag{3.62}$$

where $K = R\Gamma\Psi^T$.

3.6.3.1 Applying PDHGMu

Letting $J = \|\cdot\|_1$, $A = I$, $S = \{u : \|Ku - f\|_2 \leq \epsilon\}$ and $H(u)$ equal the indicator function $g_S(u)$ for S, application of PDHGMu yields

Algorithm: PDHGMu for Constrained l_1-Minimization

$$p^{k+1} = \Pi_{\{p:\|p\|_\infty \leq 1\}} \left(p^k + \delta_k \left((1 + \frac{\alpha_k}{\alpha_{k-1}}) u^k - \frac{\alpha_k}{\alpha_{k-1}} u^{k-1} \right) \right) \tag{3.63a}$$

$$u^{k+1} = \Pi_S \left(u^k - \alpha_k p^{k+1} \right), \tag{3.63b}$$

where $\Pi_{\{p:\|p\|_\infty \leq 1\}}$ is defined by (3.55) and

$$\Pi_S(u) = (I - K^T K)u + K^T \left(f + \frac{Ku - f}{\max\left(\frac{\|Ku-f\|_2}{\epsilon}, 1 \right)} \right)$$

thanks to the special form of K. As before, Theorem 3.4.2 applies when $\alpha_k = \alpha > 0$, $\delta_k = \delta > 0$ and $\delta \leq \frac{1}{\alpha}$. Also, since $A = I$, the case when $\delta = \frac{1}{\alpha}$ is exactly ADMM applied to ($\mathrm{SP_D}$), which is equivalent to Douglas Rachford splitting on (P).

3.6.3.2 Reversing Roles of J and H

A related approach for problem (3.62) is to apply PDHGMu with $J(u) = g_T(Ku)$ and $H(u) = \|u\|_1$, essentially reversing the roles of J and H. This will no longer satisfy the constraint at each iteration, but it does greatly simplify the projection step. The resulting algorithm is

Algorithm: PDHGRMu (reversed role version) for Constrained l_1-Minimization

$$v^{k+1} = p^k + \delta_k K\left((1+\frac{\alpha_k}{\alpha_{k-1}})u^k - \frac{\alpha_k}{\alpha_{k-1}}u^{k-1}\right) \tag{3.64a}$$

$$p^{k+1} = v^{k+1} - \delta_k \Pi_T\left(\frac{v^{k+1}}{\delta_k}\right) \tag{3.64b}$$

$$w^{k+1} = u^k - \alpha_k K^T p^{k+1} \tag{3.64c}$$

$$u^{k+1} = w^{k+1} - \alpha_k \Pi_{\{p:\|p\|_\infty \le 1\}}\left(\frac{w^{k+1}}{\alpha_k}\right). \tag{3.64d}$$

Here, v^{k+1} and w^{k+1} are just place holders and Π_T is defined by (3.56).

This variant of PDHGMu is still an application of the split inexact Uzawa method (3.30). Also, since $\|K\| \le 1$, the conditions for convergence are the same as for (3.63). Moreover, since $KK^T = I$, if $\delta = \frac{1}{\alpha}$, then this method can again be interpreted as ADMM applied to the split dual problem.

Since Π_T is much simpler to compute than Π_S, the benefit of using operator splitting to simplify the projection step is important for problems where $K^\dagger$ is not practical to deal with numerically.

3.6.4 Multiphase Segmentation

Another interesting application of PDHGMp is to the convexified multiphase segmentation model proposed in [ZGF08] and discussed in [BYT09, BCB09]. The goal is to segment a given image, $h \in \mathbb{R}^M$, into W regions where the intensities in the w^{th} region are close to given intensities $z_w \in \mathbb{R}$ and the lengths of the boundaries between regions should not be too long. This is modeled by minimizing over

$c \in \mathbb{R}^{MW}$ a functional of the form

$$g_C(c) + \sum_{w=1}^{W} \left(\|c_w\|_{TV} + \frac{\lambda}{2} \langle c_w, (h - z_w)^2 \rangle \right), \tag{3.65}$$

where

$$C = \{c = (c_1, ..., c_W) : c_w \in \mathbb{R}^M, \sum_{w=1}^{W} c_w = 1, c_w \geq 0\}$$

and g_C is the indicator function for C. This is a convex relaxation of the related nonconvex functional which additionally requires the labels, c, to only take on the values zero and one.

To apply PDHGMp, first define $\mathcal{X}_w$ to be a row selector for the c_w labels so that $\mathcal{X}_w c = c_w$. Define

$$H(c) = g_C(c) + \frac{\lambda}{2} \langle c, \sum_{w=1}^{W} \mathcal{X}_w^T (h - z_w)^2 \rangle$$

and

$$J(Ac) = \sum_{w=1}^{W} J_w(D\mathcal{X}_w c),$$

where $A = \begin{bmatrix} D\mathcal{X}_1 \\ \vdots \\ D\mathcal{X}_W \end{bmatrix}$ and

$$J_w(D\mathcal{X}_w c) = \|D\mathcal{X}_w c\|_E = \|Dc_w\|_E = \|c_w\|_{TV}.$$

Applying PDHGMp (3.35) yields

$$c^{k+1} = \Pi_C \left(c^k - \alpha \sum_{w=1}^{W} \mathcal{X}_w^T (D^T (2p_w^k - p_w^{k-1}) + \frac{\lambda}{2} (h - z_w)^2) \right)$$
$$p_w^{k+1} = \Pi_X \left(p_w^k + \delta D\mathcal{X}_w c^{k+1} \right) \qquad \text{for } w = 1, ..., W.$$

The projection Π_C can be computed with complexity of $O(MW \log(W))$ and is further discussed in Section 4.3.1.

Empirically, most of the weights $c_{m,w}$, where $m = 1, ..., M$, automatically converge to either 0 or 1. To guarantee this when visualizing the segmentation result, we estimate a binary solution $\tilde{c}^k$ such that $\tilde{c}^k_{m,w} \in \{0, 1\}$ from c^k by thresholding

$$\tilde{c}^k_{m,w} = \begin{cases} 1 & \text{if } w = \arg\max_j c^k_{m,j} \\ 0 & \text{otherwise.} \end{cases}$$

If $\arg\max_j c^k_{m,j}$ is not uniquely determined, it is chosen to be the first index where the maximum is attained.

3.7 Numerical Experiments

We perform three numerical experiments to show the modified and unmodified PDHG algorithms have similar performance and applications. The first is a comparison between PDHG, PDHGMu and ADMM applied to TV denoising. The second compares the application of PDHG and PDHGMp to a constrained TV deblurring problem. The third experiment applies PDHGMu in two different ways to a compressive sensing problem formulated as a constrained l_1 minimization problem. We also demonstrate the application of PDHGMp to the convex relaxation of multiphase segmentation discussed in Section 3.6.4.

3.7.1 Comparison of PDHGM, PDHG and ADMM for TV denoising

Here, we closely follow the numerical example presented in Table 4 of [ZC08], which compares PDHG to Chambolle's method [Cha04] and CGM [CGM99] for TV denoising. We use the same 256×256 cameraman image with intensities in $[0, 255]$. The image is corrupted with zero mean gaussian noise having standard deviation 20. We also use the same parameter $\lambda = .053$. Both adaptive and fixed stepsize strategies are compared. In all examples, we initialize $u^0 = f$ and $p^0 = 0$.

Figure 3.2: Original, noisy and benchmark denoised cameraman images

Figure 3.2 shows the clean and noisy images along with a benchmark solution for the denoised image.

Recall the PDHG algorithm for the TV denoising problem (3.36) is given by (3.9) with $K = I$. The adaptive strategy used for PDHG is the same one proposed in [ZC08] where

$$\tau_k = .2 + .008k \qquad \theta_k = \frac{.5 - \frac{5}{15+k}}{\tau_k}. \tag{3.66}$$

These can be related to the step sizes δ_k and α_k in (3.18) by

$$\delta_k = \lambda\tau_k \qquad \alpha_k = \frac{\theta_k}{\lambda(1-\theta_k)}.$$

These time steps don't satisfy the requirements of Theorem 3.5.3, which requires $\theta_k \to 1$. However, we find that the adaptive PDHG strategy (3.66), for which $\theta_k \to 0$, is better numerically for TV denoising.

When applying the PDHGMu algorithm to TV denoising, the stability requirement means using the same adaptive time steps of (3.66) can be unstable. Instead, the adaptive strategy we use for PDHGMu is

$$\alpha_k = \frac{1}{\lambda(1 + .5k)} \qquad \delta_k = \frac{1}{8.01\alpha_k} \tag{3.67}$$

Unfortunately, no adaptive strategy for PDHGMu can satisfy the requirements of Theorem 3.4.2, which assumes fixed time steps. However, the rate of convergence

of the adaptive PDHGMu strategy for TV denoising is empirically better than the fixed parameter strategies.

We also perform some experiments with fixed α and δ. A comparison is made to gradient projection (3.37). An additional comparison is made to ADMM as applied to ($\mathrm{SP_P}$). This algorithm alternates solving a Poisson equation, soft thresholding and updating the Lagrange multiplier. The explicit iterations are given by

$$\begin{aligned} u^{k+1} &= (\lambda - \delta\Delta)^{-1}(\lambda f + \delta D^T w^k - D^T p^k) \\ w^{k+1} &= \tilde{S}_{\frac{1}{\delta}}(Du^{k+1} + \frac{p^k}{\delta}) \\ p^{k+1} &= p^k + \delta(Du^{k+1} - w^{k+1}), \end{aligned} \tag{3.68}$$

where $\tilde{S}$ is defined as in (3.40) and $\Delta = -D^T D$ denotes the discrete Laplacian. This is equivalent to the split Bregman algorithm [GO09], which was compared to PDHG elsewhere in [ZC08]. However, by working with the ADMM form of the algorithm, it's easier to use the duality gap as a stopping condition since u and p have the same interpretations in both algorithms. As in [ZC08] we use the relative duality gap R for the stopping condition defined by

$$R(u,p) = \frac{F_P(u) - F_D(p)}{F_D(p)} = \frac{\left(\|u\|_{TV} + \frac{\lambda}{2}\|u-f\|_2^2\right) - \left(\frac{\lambda}{2}\|f\|_2^2 - \frac{1}{2\lambda}\|D^T p - \lambda f\|_2^2\right)}{\frac{\lambda}{2}\|f\|_2^2 - \frac{1}{2\lambda}\|D^T p - \lambda f\|_2^2},$$

which is the duality gap divided by the dual functional. The duality gap is defined to be the difference between the primal and dual functionals. This quantity is always nonnegative, and is zero if and only if (u,p) is a saddle point of (3.4) with $K = I$. Table 3.1 shows the number of iterations required for the relative duality gap to fall below tolerances of 10^{-2}, 10^{-4} and 10^{-6}. Note that the complexity of the PDHG and PDHGMu iterations scale like $O(m)$ whereas the ADMM iterations scale like $O(m \log m)$. Results for PDHGMp were identical to those for PDHGMu and are therefore not included in the table.

Algorithm	tol $= 10^{-2}$	tol $= 10^{-4}$	tol $= 10^{-6}$
PDHG (adaptive)	14	70	310
PDHGMu (adaptive)	19	92	365
PDHG $\alpha = 5, \delta = .025$	31	404	8209
PDHG $\alpha = 1, \delta = .125$	51	173	1732
PDHG $\alpha = .2, \delta = .624$	167	383	899
PDHGMu $\alpha = 5, \delta = .025$	21	394	8041
PDHGMu $\alpha = 1, \delta = .125$	38	123	1768
PDHGMu $\alpha = .2, \delta = .624$	162	355	627
PDHG $\alpha = 5, \delta = .1$	22	108	2121
PDHG $\alpha = 1, \delta = .5$	39	123	430
PDHG $\alpha = .2, \delta = 2.5$	164	363	742
PDHGMu $\alpha = 5, \delta = .1$	unstable		
PDHGMu $\alpha = 1, \delta = .5$	unstable		
PDHGMu $\alpha = .2, \delta = 2.5$	unstable		
Proj. Grad. $\delta = .0132$	48	750	15860
ADMM $\delta = .025$	17	388	7951
ADMM $\delta = .125$	22	100	1804
ADMM $\delta = .624$	97	270	569

Table 3.1: Iterations required for TV denoising

From Table 3.1, we see that PDHG and PDHGMu both benefit from adaptive stepsize schemes. The adaptive versions of these algorithms are compared in Figure 3.4(a), which plots the l_2 distance to the benchmark solution versus number of iterations. PDHG with the adaptive stepsizes outperforms all the other numerical experiments, but for identical fixed parameters, PDHGMu performed slightly better than PDHG. However, for fixed α the stability requirement, $\delta < \frac{1}{\alpha\|D\|^2}$ for PDHGMu places an upper bound on δ which is empirically about four times less than for PDHG. Table 3.1 shows that for fixed α, PDHG with larger δ outperforms PDHGMu. The stability restriction for PDHGMu is also why the same adaptive time stepping scheme used for PDHG could not be used for PDHGMu.

Table 3.1 also demonstrates that larger α is more effective when the relative duality gap is large, and smaller α is better when this duality gap is small. Since PDHG for large α is similar to projected gradient descent, roughly speaking this means the adaptive PDHG algorithm starts out closer to being gradient projection on the dual problem, but gradually becomes more like a form of subgradient descent on the primal problem.

3.7.2 Constrained TV Deblurring Example

PDHGMp and PDHG also perform similarly for constrained TV deblurring (3.59). For this example we use the same cameraman image from the previous section and let K be a convolution operator corresponding to a normalized Gaussian blur with a standard deviation of 3 in a 17 by 17 window. Letting h denote the clean image, the given data f is taken to be $f = Kh + \eta$, where η is zero mean Gaussian noise with standard deviation 1. We thus set $\epsilon = 256$. For the numerical experiments we used the fixed parameter versions of PDHG and PDHGMp

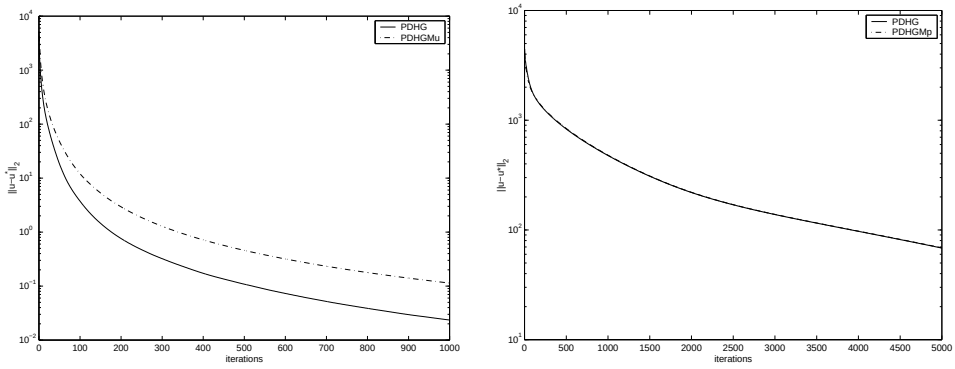
with $\alpha = .2$ and $\delta = .55$. The images h, f and the recovered image from 300 iterations of PDHGMp are shown in Figure 3.3. Figure 3.4(b) compares the l_2

Figure 3.3: Original, blurry/noisy and image recovered from 300 PDHGMp iterations

error to the benchmark solution as a function of number of iterations for PDHG and PDHGMp. Empirically, with the same fixed parameters, the performance of these two algorithms is nearly identical, and the curves are indistinguishable in Figure 3.4(b). Although many iterations are required for a high accuracy solution, Figure 3.3 shows the result can be visually satisfactory after just a few hundred iterations.

3.7.3 Constrained l_1 Minimization Examples

Here we compare PDHGMu (3.63) and the reversed role version, PDHGRMu (3.64), applied to the constrained l_1 minimization problem given by (3.62) with $\epsilon = .01$. Let $K = R\Gamma\Psi^T$, where R is a row selector, Γ is an orthogonal 2D discrete cosine transform and Ψ is an orthogonal 2D Haar wavelet transform. It follows that $KK^T = I$ and $K^\dagger = K^T$. R selects about ten percent of the DCT measurements, mostly low frequency ones. The constrained l_1 minimization model aims to recover a sparse signal in the wavelet domain that is consistent



(a) Denoising (adaptive time steps) (b) Deblurring ($\alpha = .2$, $\delta = .55$)

Figure 3.4: l_2 error versus iterations for PDHG and PDHGMp

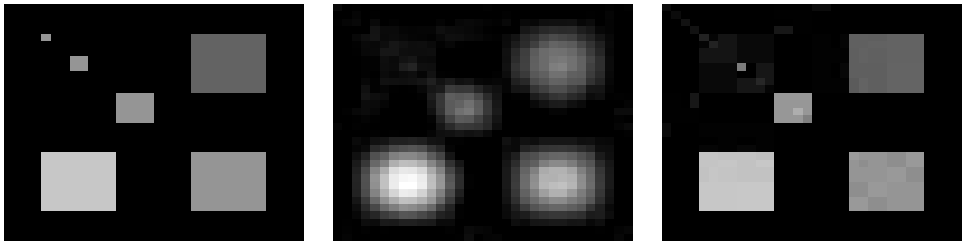


Figure 3.5: Original, damaged and benchmark recovered image

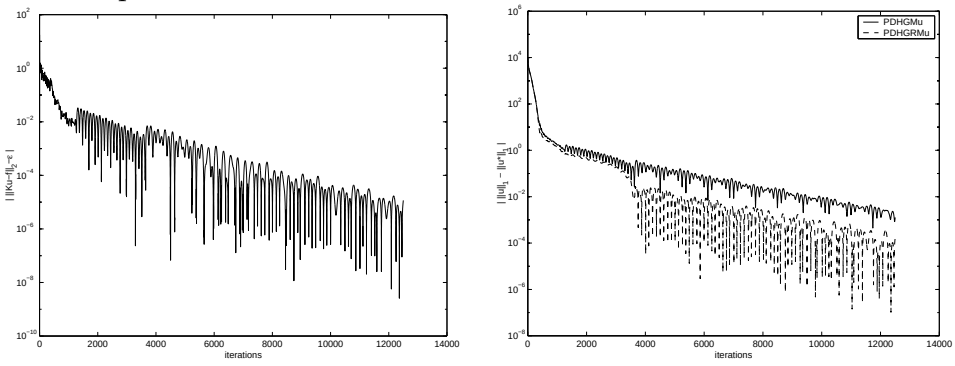
with these partial DCT measurements [CR05].

For the numerical experiments, we let $\alpha = 1$ and $\delta = 1$. Let h denote the clean image, which is a 32 by 32 synthetic image shown in figure 3.5. The data f is taken to be $R\Gamma h$. For the initialization, let $p^0 = 0$ and let $u^0 = \Psi z^0$, where $z^0 = \Gamma^T R^T R \Gamma h$ is the backprojection obtained by taking the inverse DCT of f with the missing measurements replaced by 0. Let u^* denote the solution obtained by 25000 iterations of PDHGRMu. Figure 3.5 shows h, z^0 and z^*, where $z^* = \Psi^T u^*$.

Both versions of PDHGMu applied to this problem have simple iterations that scale like $O(m)$, but they behave somewhat differently. PDHGMu (3.63) by

definition satisfies the constraint at each iteration. However, these projections onto the constraint set destroy the sparsity of the approximate solution so it can be a little slower to recover a sparse solution. PDHGRMu (3.64) on the other hand more quickly finds a sparse approximate solution but can take a long time to satisfy the constraint to a high precision.

To compare the two approaches, we compare plots of how the constraint and l_1 norm vary with iterations. Figure 3.6(a) plots $|\|Ku^k - f\|_2 - \epsilon|$ against the iterations k for PDHGRMu. Note this is always zero for PDHGMu, which stays on the constraint set. Figure 3.6(b) compares the differences $|\|u^k\|_1 - \|u^*\|_1|$ for both algorithms on a semilog plot, where $\|u^*\|_1$ is the l_1 norm of the benchmark solution. The empirical rate of convergence to $\|u^*\|_1$ was similar for both algorithms despite the many oscillations. PDHGRMu was a little faster to recover a sparse solution, but PDHGMu has the advantage of staying on the constraint set. For different applications with more complicated K, the simpler projection step for PDHGRMu would be an advantage of that approach.



(a) Constraint versus iterations for PDHGRMu ($\alpha = 1$, $\delta = 1$)

(b) l_1 Norm Comparison ($\alpha = 1$, $\delta = 1$)

Figure 3.6: Comparison of PDHGRMu and PDHGMu

3.7.4 Multiphase Segmentation Example

Numerical experiments for the convex relaxed multiphase segmentation model (3.65) are performed in [BYT09], where an expectation maximization algorithm is applied after forming a smooth approximation to the dual problem. As discussed in Section 3.6.4, it's also possible to directly apply PDHGMp without altering the functional. This is tested on the problem of segmenting a brain scan image into five regions. In terms of the functional (3.65), h is the original image shown in Figure 3.7, $\lambda = .0025$ and we let

$$z = \begin{bmatrix} 75 & 105 & 142 & 178 & 180 \end{bmatrix}.$$

The time step parameters α and δ are each set to $\frac{.995}{\sqrt{40}}$. The result, which is also shown in Figure 3.7, is obtained by thresholding c^{1000} and visualized by setting the intensities to z on the segmented regions.

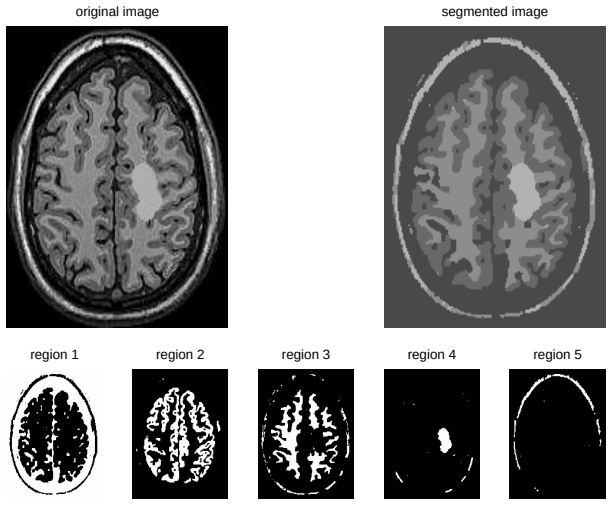


Figure 3.7: Segmentation of brain image into 5 regions

It is possible to improve the regularization on the length of the boundaries by introducing μ_w parameters in front of the $\|c_w\|_{TV}$ terms and adjusting them

according to any a-priori assumptions we can make about the shapes of the desired regions.

If z is not known in advance, one can additionally minimize over z. The resulting functional is nonconvex, but a practical method for approximating its minimizers is to follow the approach of Chan-Vese segmentation [CV01] and alternate minimization of c with z fixed and minimization of z with c fixed, which leads to weighted average updates for the z_w.

CHAPTER 4

A Convex Model for Image Registration

4.1 Introduction

In this chapter, we propose a convex model for image registration. The model uses a graph-based formulation and minimizes a convex function on the edges of the graph instead of working directly with the displacement field. A classical approach for registering given images u, $\phi : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ is to minimize the Horn-Schunck model, [HS81]

$$\frac{1}{2}\|\phi(x + v(x)) - u(x)\|^2 + \frac{\gamma}{2}\|\nabla v_1\|^2 + \frac{\gamma}{2}\|\nabla v_2\|^2, \tag{4.1}$$

with respect to the displacement field v. Instead of seeking a global minimum of this nonconvex functional, we reformulate it as a convex minimization problem, but without linearizing the fidelity term, which would require a small deformation assumption, and also while avoiding the interpolation difficulty that arises when discretizing $\phi(x + v)$. Rather than discretizing a continuous model, we will work in a discrete setting throughout. Given images $u \in \mathbb{R}^{m_r \times m_c}$ and $\phi \in \mathbb{R}^{n_r \times n_c}$, consider defining a graph as in Figure 4.1. The nodes correspond to pixel centers in u and ϕ, and the edges are defined to connect each pixel in u to a neighborhood of pixels in ϕ. The unknown edge weights will correspond to the coefficients of a weighted average. The main idea behind the convex reformulation is then to replace $\phi(x+v)$, v_1 and v_2 in (4.1) with weighted averages of intensities and pixel

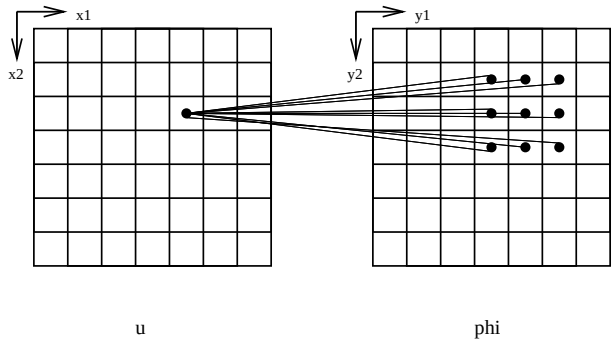


Figure 4.1: Construction of edges $e_{i,j}$

locations. The discrete interpolation corresponding to $\phi(x+v)$ will be a weighted average of intensities of ϕ with the weights corresponding to edge weights on the graph. Similarly the displacement will be modeled as the difference between the pixel locations in u and the same weighted averages of the pixel locations in ϕ.

A crucial requirement for the weighted average approach to be justified is that the weights be localized. For the interpolation to make sense, it should only depend on nearby pixels. Therefore the edge weights should be zero outside a small neighborhood around the weighted average of the ϕ pixel locations. Directly enforcing this would unfortunately correspond to a nonconvex constraint. However, it is possible to indirectly encourage the weights to cluster by adding a convex term to the functional requiring the weights to be spatially smooth. Details are in the following section.

There are many other approaches that aim to convexify or at least partially convexify Horn-Schunck related models for image registration. Many authors make use of a multiscale approach, working from coarse to fine images, since applying low pass filters to the images can make the energy more convex [LC01]. As already mentioned, one can obtain a convex approximation to (4.1) by linearizing $\frac{1}{2}\|\phi(x+v(x))-u(x)\|^2$ to obtain $\frac{1}{2}\|\phi(x)+\langle\nabla\phi(x),v(x)\rangle-u(x)\|_2^2$. A multiscale

approach can be used to get around the drawback that the linearization is only valid for small deformations. It's shown in [BBP04] that applying a coarse to fine strategy to either the original nonconvex functional or to the linearized version amounts to essentially the same thing. A coarse to fine approach is also used in [ZPB07] in a real time algorithm for computing optical flow. Another very interesting approach is the discrete functional lifting method in [GBO09], which recovers a global minimum of the nonconvex functional by solving an equivalent convex problem. It is related to the discrete and continuous functional lifting approaches of [Ish03] and [PSG08].

The organization of this chapter is as follows. In Section 4.2, the weighted average based convex model for image registration is defined and discussed. Section 4.3 explains how a variant of the primal dual hybrid gradient (PDHG) method [ZC08, EZC09] can be used to minimize the resulting functional. Numerical results for several registration examples are presented in Section 4.4. Section 4.5 discuses extensions of the convex model such as incorporating l_1 and TV terms into the functional, modifications of the numerical scheme as well as other applications of the numerical approach to similar models.

4.2 Formulation of Convex Registration Model

Let the images $u \in \mathbb{R}^{m_r \times m_c}$ and $\phi \in \mathbb{R}^{n_r \times n_c}$ be given. Assume that the pixel intensities are not changed by the optimal displacement. This is the key assumption behind the data fidelity term in (4.1). Also assume we are given a guess $\nu = (\nu_1, \nu_2)$ of the displacement field and upper bounds $r_1, r_2 \geq 0$ such that if v^* is the true displacement, then $\|\nu_1 - v_1^*\|_\infty \leq r_1$ and $\|\nu_2 - v_2^*\|_\infty \leq r_2$. The images are allowed to be at different resolutions, but we assume that the dimensions of the pixels are known and are such that there are no major scale differences be-

tween the two images. Let $M = m_r m_c$ be the number of pixels in u and $N = n_r n_c$ the number of pixels in ϕ. Consider a graph $G(\mathcal{V}, \mathcal{E})$ with a node for each pixel in u and ϕ. We can write $\mathcal{V}$ as $\mathcal{V} = \mathcal{V}_u \bigcup \mathcal{V}_\phi$ where $\mathcal{V}_\phi = \{1, ..., N\}$ indexes the nodes from ϕ and $\mathcal{V}_u = \{N+1, ..., M+N\}$ indexes the nodes from u. We will consider $u \in \mathbb{R}^M$ to be a function on the nodes in $\mathcal{V}_u$ and similarly $\phi \in \mathbb{R}^N$ a function on the nodes in $\mathcal{V}_\phi$. Also define $x_1,\ x_2 \in \mathbb{R}^M$ to be functions on nodes in V_u, and similarly define y_1 and y_2 to be functions on the nodes in V_ϕ such that (x_1^i, x_2^i) is the location of the center of the pixel corresponding to node $i \in V_u$, and (y_1^j, y_2^j) is the location of the center of the pixel corresponding to node $j \in V_\phi$. Now let there be an edge between node $i \in \mathcal{V}_u$ and node $j \in V_\phi$ if $|y_1^j - (x_1^i + \nu_1^i)| \leq r_1$ and $|y_2^j - (x_2^i + \nu_2^i)| \leq r_2$. Denote the total number of edges by e. Figure 4.1 illustrates how these edges connect each node i to a group of nodes j.

Note that if r is too large, the number of edges defined could be so large as to cause memory problems in a numerical implementation. One way of avoiding this is to limit the size of the allowed deformation. Some alternatives include only defining edges on a subset of the nodes satisfying the displacement bounds, using a coarse to fine multiscale approach, or designing a transformation that gives a lower dimensional approximation of the edge weights. Here, we will make use of the multiscale approach in numerical implementations.

Assume for simplicity that any $(w_1, w_2) \in \mathbb{R}^2$ satisfying $|w_1 - (x_1^i + \nu_1^i)| \leq r_1$ and $|w_2 - (x_2^i + \nu_2^i)| \leq r_2$ for some $i \in \mathcal{V}_u$ also satisfies $\min_j(y_1^j) \leq w_1 \leq \max_j(y_1^j)$ and $\min_j(y_2^j) \leq w_2 \leq \max_j(y_2^j)$ (ϕ can be padded if this doesn't hold). Let $c \in \mathbb{R}^e$ be a function on the edges such that $c_{i,j} \geq 0$ and $\sum_{j \sim i} c_{i,j} = 1$ for each $i \in \mathcal{V}_u$. Now we model the displacement v to be the difference between the c-weighted

average of y and x.

$$(v_1^i, v_2^i) = \left((\sum_{j\sim i} c_{i,j} y_1^j - x_1^i), (\sum_{j\sim i} c_{i,j} y_2^j - x_2^i) \right) \tag{4.2}$$

These weighted averages can be represented more compactly in terms of the edge-node adjacency matrix $[Q\ R] \in \mathbb{R}^{e\times(M+N)}$ for the graph G, where Q corresponds to the nodes in V_ϕ and R corresponds to the nodes in V_u. For each edge $e_{i,j}$ where $i \in \mathcal{V}_u$ and $j \in \mathcal{V}_\phi$, $Q_{e,j} = -1$ and $R_{e,i} = 1$. All other entries of Q and R equal zero. Let $\operatorname{diag}(c)$ denote the diagonal matrix with the vector c along the diagonal. The operation of taking c-weighted averages of a function on $\mathcal{V}_\phi$ for each node $i \in \mathcal{V}_u$ can be represented in matrix notation by $-(R^T \operatorname{diag}(c) R)^{-1} R^T \operatorname{diag}(c) Q$. The constraint that the weights on the edges coming out of each node i sum to 1 can be written as $R^T c = 1$ or $R^T \operatorname{diag}(c) R = I$. Thus the c-weighted averages of ϕ can be written as $-R^T \operatorname{diag}(c) Q\phi$. Now define

$$A_\phi = -R^T \operatorname{diag}(Q\phi).$$

This means $A_\phi c$ represents the c-weighted averages of ϕ. Similarly, define

$$A_{y_1} = -R^T \operatorname{diag}(Q y_1)$$

and

$$A_{y_2} = -R^T \operatorname{diag}(Q y_2).$$

$A_{y_1} c$ and $A_{y_2} c$ represent the c-weighted averages of y_1 and y_2 respectively. In terms of these matrices, the displacements $v_1, v_2 \in \mathbb{R}^M$ are modeled by

$$v_1 = A_{y_1} c - x_1 \qquad , \qquad v_2 = A_{y_2} c - x_2.$$

Let C denote the convex set that the weights c are constrained to lie in. This set is defined by

$$C = \left\{ c \in \mathbb{R}^e : c_{i,j} \geq 0 \qquad \text{and} \qquad \sum_{j\sim i} c_{i,j} = 1 \right\}. \tag{4.3}$$

Let g_C be the indicator function for C defined by

$$g_C(c) = \begin{cases} 0 & \text{if } c \in C \\ \infty & \text{otherwise} \end{cases} . \tag{4.4}$$

We also need to define the discretized gradient D that will act on vectorized images in $\mathbb{R}^M$. The convention for vectorizing a m_r by m_c matrix will be to stack the columns so that the (r, c) element of the matrix corresponds to the $(c-1)m_r+r$ element of the vector. Consider a new graph, $G_D(\mathcal{V}_u, \mathcal{E}_D)$ with nodes in $\mathcal{V}_u$ and edges that correspond to forward differences as is shown in Figure 4.2. Index the

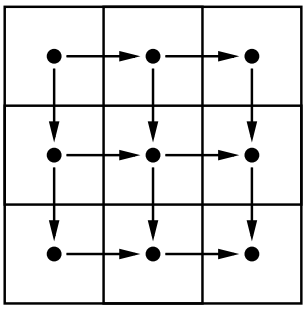


Figure 4.2: Graph for defining D

M nodes by $(c-1)m_r + r$ and the $e_D = 2m_r m_c - m_r - m_c$ edges arbitrarily. Now define $D \in \mathbb{R}^{e_D \times M}$ to be the edge node adjacency matrix for the graph G_D. For each edge ξ with endpoint indices (i, j), $i < j$, define

$$D_{\xi,k} = \begin{cases} -1 & \text{for } k = i \\ 1 & \text{for } k = j \\ 0 & \text{for } k \neq i, j \end{cases} . \tag{4.5}$$

The matrix D thus defined can be interpreted as the discrete gradient, and likewise $-D^T$ represents the discrete divergence operator. The graph definition implicitly assumes Neumann boundary conditions.

With this notation we can define the weighted average analogue to (4.1),

$$\frac{1}{2}\|A_\phi c - u\|_2^2 + \frac{\eta}{2}\|D(A_{y_1}c - x_1)\|_2^2 + \frac{\eta}{2}\|D(A_{y_2}c - x_2)\|_2^2 + g_C(c). \tag{4.6}$$

We will assume that the displacement is smooth and therefore keep the regularization terms that penalize the l_2 norm squared of its gradient. It will be convenient for notational reasons to define

$$R_1(z) = \frac{\eta}{2}\|D(z - x_1)\|_2^2$$

and

$$R_2(z) = \frac{\eta}{2}\|D(z - x_2)\|_2^2$$

so that these regularization terms can be rewritten as $R_1(A_{y_1}c) + R_2(A_{y_2}c)$. For applications where discontinuities are expected in the displacement field, total variation regularization terms could be used here instead.

Since the displacement is assumed to be smooth, the weights themselves should also be spatially smooth. Moreover enforcing this should encourage the weights to be localized, which needs to happen for the model of the displacement to make sense. For simplicity, assume every node $i \in \mathcal{V}_u$ has the same number of edges connecting it to nodes in $\mathcal{V}_\phi$ and moreover that those nodes in $\mathcal{V}_\phi$ form a rectangle of dimension w_r by w_c that will be referred to as the search window. To be consistent with the bounds r_1 and r_2, we can take $w_r = 2r_2 + 1$ and $w_c = 2r_1 + 1$. There are $W = w_r w_c$ weights for each node in $\mathcal{V}_u$. Note that the total number of edge weights is $e = MW$. Let w be an index for the weights in the search window and define $\mathcal{X}_w \in \mathbb{R}^{M \times e}$ to be a row selector ($\mathcal{X}_w \mathcal{X}_w^T = I$) for the w^{th} weight. So $\mathcal{X}_w$ applied to c returns a vector of just those M weights that correspond to the index w. Now we can encourage spatial smoothness of the weights by adding

$$\sum_{w=1}^{W} \|D\mathcal{X}_w c\|_2$$

to the functional. It will later be helpful to rewrite this in terms of an indicator function g_B for the unit l_2 ball

$$B = \{p : \|p\|_2 \leq 1\}$$

because the numerical approach will involve projecting onto this set. Since the Legendre transform of a norm can be interpreted as the indicator function for the unit ball in the dual norm,

$$\|D\mathcal{X}_w c\|_2 = g_B^*(D\mathcal{X}_w c),$$

where g_B^* denotes the Legendre transform of g_B.

The quadratic data fidelity term doesn't robustly handle outliers. To remedy this, we will replace the quadratic term with convex constraints that control both the local error and the average error for $A_\phi c - u$. To control the local error, we can require that $\|\frac{A_\phi c - u}{\tau}\|_\infty \leq 1$ for some data dependent $\tau \in \mathbb{R}^M$ and where the division by τ is understood to be componentwise. To control the average error we can require that $\|A_\phi c - u\|_2 \leq \epsilon$ for some $\epsilon \geq 0$. These constraints can be added to the functional as indicator functions for the appropriate convex sets. Let

$$T_2 = \{z : \|z - u\|_2 \leq \epsilon\}$$

and let

$$T_\infty = \{z : \|\frac{z - u}{\tau}\|_\infty \leq 1\}.$$

Let g_{T_2} and g_{T_∞} be the indicator functions for T_2 and T_∞. Another possibility is to use the l_1 norm for the data fidelity term, which is considered in Section 4.5.1.

Altogether, the proposed convex functional for image registration is given by

$$F(c) = g_C(c) + \sum_{w=1}^{W} g_B^*(D\mathcal{X}_w c) + g_{T_2}(A_\phi c) + g_{T_\infty}(A_\phi c) + R_1(A_{y_1} c) + R_2(A_{y_2} c). \quad (4.7)$$

A minimizer exists as long as the set

$$\{c : c \in C, \|A_\phi c - u\|_2 \leq \epsilon, \|\frac{A_\phi c - u}{\tau}\|_\infty \leq 1\}$$

is nonempty.

4.3 Numerical Approach

To solve (4.7) we will use the PDHGMp variant of the PDHG algorithm, which is defined by (3.35). It's application to (4.7) will make use of the operator splitting techniques discussed in Section 3.6.1.

4.3.1 Application of PDHGMp

Since F (4.7) is in the form of (3.52), the PDHGMp method can be directly applied. However, the relative scaling of the matrices $D\mathcal{X}_w$, A_ϕ, A_{y_1} and A_{y_2} can affect the numerical performance. We therefore introduce scaling factors s_w, s_{ϕ_2}, s_{ϕ_∞}, s_{y_1} and s_{y_2} and define

$$\begin{aligned}
\tilde{g}_B^*(z_w) &= g_B^*(s_w z_w) \\
\tilde{g}_{T_2}(z_{\phi_2}) &= g_{T_2}(s_{\phi_2} z_{\phi_2}) \\
\tilde{g}_{T_\infty}(z_{\phi_\infty}) &= g_{T_\infty}(s_{\phi_\infty} z_{\phi_\infty}) \\
\tilde{R}_1(z_{y_1}) &= R_1(s_{y_1} z_{y_1}) \\
\tilde{R}_2(z_{y_2}) &= R_2(s_{y_2} z_{y_2})
\end{aligned}$$

so that F can be equivalently written as

$$F(c) = g_C(c) + \sum_{w=1}^{W} \tilde{g}_B^*(\frac{D\mathcal{X}_w c}{s_w}) + \tilde{g}_{T_2}(\frac{A_\phi c}{s_{\phi_2}}) + \tilde{g}_{T_\infty}(\frac{A_\phi c}{s_{\phi_\infty}}) + \tilde{R}_1(\frac{A_{y_1} c}{s_{y_1}}) + \tilde{R}_2(\frac{A_{y_2} c}{s_{y_2}}). \tag{4.8}$$

To apply PDHGMp, let

$$A = \begin{bmatrix} \frac{D\mathcal{X}_1}{s_1} \\ \vdots \\ \frac{D\mathcal{X}_w}{s_W} \\ \frac{A_\phi}{s_\phi} \\ \frac{A_\phi}{s_\phi} \\ \frac{A_{y_1}}{s_{y_1}} \\ \frac{A_{y_2}}{s_{y_2}} \end{bmatrix}, \qquad p = \begin{bmatrix} p_1 \\ \vdots \\ p_W \\ p_{\phi_2} \\ p_{\phi_\infty} \\ p_{y_1} \\ p_{y_2} \end{bmatrix}, \qquad H(c) = g_C(c)$$

and

$$J(Ac) = \sum_{w=1}^{W} \tilde{g}_B^*(\frac{D\mathcal{X}_w c}{s_w}) + \tilde{g}_{T_2}(\frac{A_\phi c}{s_{\phi_2}}) + \tilde{g}_{T_\infty}(\frac{A_\phi c}{s_{\phi_\infty}}) + \tilde{R}_1(\frac{A_{y_1} c}{s_{y_1}}) + \tilde{R}_2(\frac{A_{y_2} c}{s_{y_2}}).$$

The initialization is arbitrary. One iteration of PDHGMp applied to $F(c)$ and including the scale factors consists of the following minimization steps:

$$\begin{aligned} c^{k+1} = \arg\min_c g_C(c) + \frac{1}{2\alpha} \Bigg\| c - \Bigg(c^k - \alpha \sum_{w=1}^{W} \frac{\mathcal{X}_w^T D^T (2p_w^k - p_w^{k-1})}{s_w} \\ - \alpha \frac{A_\phi^T (2p_{\phi_2}^k - p_{\phi_2}^{k-1})}{s_{\phi_2}} - \alpha \frac{A_\phi^T (2p_{\phi_\infty}^k - p_{\phi_\infty}^{k-1})}{s_{\phi_\infty}} \\ - \alpha \frac{A_{y_1}^T (2p_{y_1}^k - p_{y_1}^{k-1})}{s_{y_1}} - \alpha \frac{A_{y_2}^T (2p_{y_2}^k - p_{y_2}^{k-1})}{s_{y_2}} \Bigg) \Bigg\|_2^2 \end{aligned}$$

$$p_w^{k+1} = s_w \arg\min_{p_w} g_B(p_w) + \frac{s_w^2}{2\delta} \left\| p_w - \frac{(p_w^k + \frac{\delta D \mathcal{X}_w c^{k+1}}{s_w})}{s_w} \right\|_2^2 \qquad \text{for } w = 1, ..., W$$

$$p_{\phi_2}^{k+1} = s_{\phi_2} \arg\min_{p_{\phi_2}} g_{T_2}^*(p_{\phi_2}) + \frac{s_{\phi_2}^2}{2\delta} \left\| p_{\phi_2} - \frac{(p_{\phi_2}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_2}})}{s_{\phi_2}} \right\|_2^2$$

$$p_{\phi_\infty}^{k+1} = s_{\phi_\infty} \arg\min_{p_{\phi_\infty}} g_{T_\infty}^*(p_{\phi_\infty}) + \frac{s_{\phi_\infty}^2}{2\delta} \left\| p_{\phi_\infty} - \frac{(p_{\phi_\infty}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_\infty}})}{s_{\phi_\infty}} \right\|_2^2$$

$$p_{y_1}^{k+1} = s_{y_1} \arg\min_{p_{y_1}} R_1^*(p_{y_1}) + \frac{s_{y_1}^2}{2\delta} \left\| p_{y_1} - \frac{(p_{y_1}^k + \frac{\delta A_{y_1} c^{k+1}}{s_{y_1}})}{s_{y_1}} \right\|_2^2$$

$$p_{y_2}^{k+1} = s_{y_2} \arg\min_{p_{y_2}} R_2^*(p_{y_2}) + \frac{s_{y_2}^2}{2\delta} \left\| p_{y_2} - \frac{(p_{y_2}^k + \frac{\delta A_{y_2} c^{k+1}}{s_{y_2}})}{s_{y_2}} \right\|_2^2$$

There are simple to compute, explicit formulas for each of the minimization steps. A helpful tool for writing down some of the formulas is the general Moreau decomposition (2.3.1).

Substituting formulas for the minimization steps and using the Moreau decomposition to simplify the last four updates, the PDHGMp method applied to

$F(c)$, still including the scale factors, is to iterate

$$c^{k+1} = \Pi_C \left(c^k - \alpha \sum_{w=1}^{W} \frac{\mathcal{X}_w^T D^T (2p_w^k - p_w^{k-1})}{s_w} - \alpha \frac{A_\phi^T (2p_{\phi_2}^k - p_{\phi_2}^{k-1})}{s_{\phi_2}} \right.$$
$$\left. -\alpha \frac{A_\phi^T (2p_{\phi_\infty}^k - p_{\phi_\infty}^{k-1})}{s_{\phi_\infty}} - \alpha \frac{A_{y_1}^T (2p_{y_1}^k - p_{y_1}^{k-1})}{s_{y_1}} - \alpha \frac{A_{y_2}^T (2p_{y_2}^k - p_{y_2}^{k-1})}{s_{y_2}} \right)$$

$$p_w^{k+1} = s_w \Pi_B \left(\frac{(p_w^k + \frac{\delta D \mathcal{X}_w c^{k+1}}{s_w})}{s_w} \right) \qquad \text{for } w = 1, ..., W$$

$$p_{\phi_2}^{k+1} = p_{\phi_2}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_2}} - \frac{\delta}{s_{\phi_2}} \Pi_{T_2} \left((p_{\phi_2}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_2}}) \frac{s_{\phi_2}}{\delta} \right)$$

$$p_{\phi_\infty}^{k+1} = p_{\phi_\infty}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_\infty}} - \frac{\delta}{s_{\phi_\infty}} \Pi_{T_\infty} \left((p_{\phi_\infty}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_\infty}}) \frac{s_{\phi_\infty}}{\delta} \right)$$

$$p_{y_1}^{k+1} = p_{y_1}^k + \frac{\delta A_{y_1} c^{k+1}}{s_{y_1}} - (I + \frac{\eta s_{y_1}^2}{\delta} D^T D)^{-1} \left(\eta s_{y_1} D^T D x_1 + p_{y_1}^k + \frac{\delta A_{y_1} c^{k+1}}{s_{y_1}} \right)$$

$$p_{y_2}^{k+1} = p_{y_2}^k + \frac{\delta A_{y_2} c^{k+1}}{s_{y_2}} - (I + \frac{\eta s_{y_2}^2}{\delta} D^T D)^{-1} \left(\eta s_{y_2} D^T D x_2 + p_{y_2}^k + \frac{\delta A_{y_2} c^{k+1}}{s_{y_2}} \right)$$

Here Π_C, Π_B, Π_{T_2} and Π_{T_∞} denote the orthogonal projections onto the convex sets C, B, T_2 and T_∞ respectively. Formulas for the latter three are

$$\Pi_B(p) = \frac{p}{\max(\|p\|_2, 1)},$$

$$\Pi_{T_2}(z) = u + \frac{z - u}{\max(\frac{\|z-u\|_2}{\epsilon}, 1)}$$

and

$$\Pi_{T_\infty}(z) = u + \frac{z - u}{\max(\frac{|z-u|}{\tau}, 1)},$$

where for Π_{T_∞} the max and division are understood in a componentwise sense. Although there isn't a formula for $\Pi_C(c)$, it can still be computed efficiently with complexity of $O(MW \log(W))$. In describing this projection, it is helpful to reindex c. Computing $\Pi_C(c)$, where $c \in \mathbb{R}^{MW}$, amounts to orthogonally projecting M vectors in $\mathbb{R}^W$ onto the positive face of the l_1 unit ball in $\mathbb{R}^W$. Let $c_m \in \mathbb{R}^W$ for $m = 1, ..., M$ denote those vectors. Then the elements $c_{m,w}$ of c_m must project

either to zero or to $c_{m,w} - \theta_m$, where θ_m is such that $\sum_{w=1}^{W} c_{m,w} = 1$ and $c_{m,w} \geq 0$. Therefore the projection can be computed once the thresholds θ_m are found, which can be done by sorting and using a bisection strategy to determine how many elements project to zero. Finally note that $D^T D$ denotes minus the discrete Laplacian corresponding to Neumann boundary conditions. The corresponding discrete Poisson equation can be efficiently solved with $O(M \log(M))$ complexity using the discrete cosine transform.

4.3.2 Discussion of Parameters

There are some necessary conditions on the parameters. The parameters $\epsilon \in \mathbb{R}$ and $\tau \in \mathbb{R}^M$, which appear in the definition of the sets T_2 and T_∞, must be large enough so that a minimizer exists. Also, α and δ must be positive and satisfy $\alpha\delta < \frac{1}{\|A\|^2}$. Adjusting ϵ and τ changes the underlying model and will affect the solution. Changing α and δ, as long as they satisfy the stability requirement, only affects the rate of convergence.

The scaling factors s_w, s_{ϕ_2}, s_{ϕ_∞}, s_{y_1} and s_{y_2} also affect the rate of convergence but mostly they alter the relative weight that each term of the functional has on each iteration. Adjusting these factors doesn't change the model or the eventual solution, but it can for example make the numerical solution satisfy the data constraint in fewer iterations at the cost of it taking more iterations for the weights to become smooth. Or vice versa, the scaling factors can encourage early iterates to be smooth at the cost of more iterations being needed to satisfy the data fidelity constraints. A natural approach to defining the scaling parameters is to try to give the five terms (thinking of the sum over w as a single term) in $J(Ac)$ roughly equal weight.

The parameter η does affect the model in the sense of altering the relative

importance of the smooth displacement regularizer and the smooth weights regularizer. If η is too large, then a smooth displacement might come at the cost of having insufficiently smooth and therefore possibly nonlocal weights. If η is too small, the weights themselves will be spatially smooth, but although the smoothness of the displacement is indeed encouraged by smoothness of the weights, it is not always sufficiently enforced that way. Moreover, it empirically takes many more iterations to get a reasonable solution when η is too small. For some examples, choosing η too small can again result in nonlocal weights. Since the displacement model assumes local weights, it's therefore important to choose η well to avoid errors in the registration.

4.3.3 Multiscale Approach

A downside of the model (4.7) is the large number of variables involved. Although it's a convex registration model that allows for large deformations, the number of edge weights in the graph formulation can be impractically large. By practical necessity, a coarse to fine multiscale approach is used for the numerical examples in Section 4.4. A dimension reduction idea is mentioned in Section 4.5.4 but not implemented.

The multiscale approach works by downsampling the original images by a factor of two as many times as is necessary for the number of pixels within the maximum displacement estimate not to be impractically large. The effect of one level of downsampling is illustrated in Figure 4.3. The convex registration problem is first solved for the low resolution images. Then the resulting displacement solution given by $(A_{y_1}c - x_1, A_{y_2}c - x_2)$ is upsampled by a factor of two and used as an initial guess when solving the convex registration problem for the next finer resolution. If one assumes the global coarse solution is close, say within half a

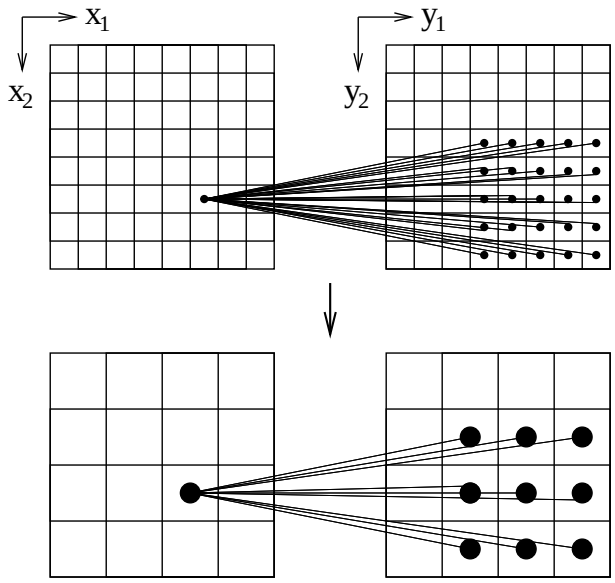


Figure 4.3: Effect of downsampling on resolution and search window size

coarse pixel, to the global fine solution, then small search windows suffice for all the successive applications of the method to the finer resolution images. For example, after the coarse problem is solved, a three by three window of weights could be used for the remaining problems.

An advantage of the multiscale approach is that by downsampling enough times, the size of the search window of weights can be made small enough to automatically satisfy the localized weights assumption. And the number of extra variables can be made small enough so that the computation remains efficient. However, the coarse solutions aren't guaranteed to be close to the true global minimum of the functional at the fine scale. In fact, downsampling too many times will result in poor solutions. So it is best to employ the multiscale strategy as little as possible, only as much as is practically necessary to achieve reasonable computation times.

4.4 Numerical Examples

In this section, the performance of the model and numerical approach is illustrated for three examples.

The first example is a synthetic image of the letter E which is to be registered with a translated, rotated version. See Figure 4.4. This example illustrates that the model succeeds in filling in the large homogeneous regions with a smooth displacement field.

The second example is low resolution digital photo of two pencils on a desk which is to be registered with a version that has undergone a large translation and also some rotation. See Figure 4.5. The initial displacement is chosen to align the rightmost pencil in u with the leftmost pencil in ϕ. This would be a local minimum for the classical registration model (4.1), but the global solution of the convex model correctly registers the images despite the large translation and challenging initialization. The pencil example will also be used to illustrate the problems that occur if η is too small or if the fidelity terms are too weak. These both result in nonlocal weights and a poor solution, whereas with well chosen parameters the weights do indeed localize and the solution is good.

The third example is a brain scan where a section in the middle has been deformed, leaving the outer regions unchanged. See Figure 4.6. This example is from [TGS06, LGD09] and the ground truth is known. A comparison with the ground truth displacement is plotted in Figure 4.7. In addition to showing that the model can successfully register this medical image, the example is also used to illustrate the need for the smoothing regularizer on the weights. If η is chosen too large, the displacement should still of course be smooth but the weights themselves may not be smooth enough. It's then possible for nonlocal

weights to conspire to satisfy the data fidelity constraints and yield a smooth displacement while still giving the wrong solution.

4.4.1 Parameter Definitions

Most of the parameters are chosen similarly for the numerical examples. The parameter η that balances the smooth displacement and smooth weights regularizers is usually chosen to be 1 except for the examples that illustrate what goes wrong when η is too small or too large. The weights $\tau \in \mathbb{R}^m$ for the l_∞ data constraint are data dependent, small in homogeneous regions and large near discontinuities. For all the examples τ_i is defined by taking the difference of the maximum and minimum intensities in the three by three neighborhood around the i^{th} pixel, multiplying by .75 and adding .5. This ensures $|A_\phi c - u|$ is never much larger that could be expected from interpolation errors. Recall that $A_\phi c - u$ is the difference between the c-weighted averages of ϕ and u. For the l_2 data fidelity constraint, the best choice of ϵ depends on the problem. The scaling parameters are designed to normalize the matrices $\sum_{w=1}^{W} \mathcal{X}_w^T D^T$, A_ϕ^T, $A_{y_1}^T$ and $A_{y_2}^T$ so that they have roughly the same operator norms. For all the numerical examples, define

$$\begin{aligned} s_w &= 2\sqrt{10W} \\ s_{\phi_2} &= \|\phi\|_\infty \sqrt{5W} \\ s_{\phi_\infty} &= s_{\phi_2} \\ s_{y_1} &= n_c \Delta_{y_c} \sqrt{5W} \\ s_{y_2} &= n_r \Delta_{y_r} \sqrt{5W}, \end{aligned}$$

where Δ_{y_c} and Δ_{y_r} denote the dimensions of a single pixel in ϕ. The choice of numerical parameters α and δ can greatly affect the rate of convergence. It is

best for $\alpha\delta$ to be close to the stability bound $\frac{1}{\|A\|^2}$. An upper bound, a for $\|A\|^2$ is

$$a = \frac{8W}{s_w^2} + \|\frac{A_\phi A_\phi^T}{s_{\phi_2}^2} + \frac{A_\phi A_\phi^T}{s_{\phi_\infty}^2} + \frac{A_{y_1} A_{y_1}^T}{s_{y_1}^2} + \frac{A_{y_2} A_{y_2}^T}{s_{y_2}^2}\|,$$

which is straightforward to compute because $A_\phi A_\phi^T$, $A_{y_1} A_{y_1}^T$ and $A_{y_2} A_{y_2}^T$ are diagonal matrices. Reasonable choices for α and δ are $\alpha = \frac{.995}{s_{\phi_2}\sqrt{a}}$ and $\delta = \frac{.995 s_{\phi_2}}{\sqrt{a}}$.

4.4.2 Multiscale Implementation and Stopping Condition

To speed up the numerical implementation, a coarse to fine multiscale approach as described in 4.3.3 is used for all the following examples. The letter E and pencil examples are both initially downsampled twice. The brain example is initially downsampled three times. The downsampling is always by a factor of two. Once a coarse solution is obtained and the displacement computed, bilinear interpolation is used on the coarse displacement field to obtain an initial guess for the next finer resolution. Since the coarse solution is assumed to be close to the true solution, small search windows, (usually 3×3 or 5×5), are used for all the finer resolution registration problems with good initial displacement estimates.

The stopping condition used for the E and pencil examples is

$$\|c^{k+1} - c^k\|_\infty \leq \frac{.002}{W}, \tag{4.9}$$

where W is the total number of weights in each search window. To speed up computation for the brain example, .002 was replaced by .004.

4.4.3 Results

For the letter E example, Figure 4.4 shows the original 128×128 images u and ϕ, the c-weighted averages of ϕ and the computed displacement. The parameters used are $\alpha = \frac{.995}{s_{\phi_2}\sqrt{a}}$, $\delta = \frac{.995 s_{\phi_2}}{\sqrt{a}}$, $\epsilon = \frac{\sqrt{M}}{2}$ and $\eta = 1$. The maximum displacement

scale	iterations
2	3890
1	2348
0	1552

Table 4.1: Iterations required for registering E example

(r_1, r_2) is set to $(16, 8)$. For all scales beyond the coarsest, three by three search windows are used. The number of iterations required at each scale to satisfy the stopping condition (4.9) is given in Table 4.1, where scale refers to the number of downsamplings.

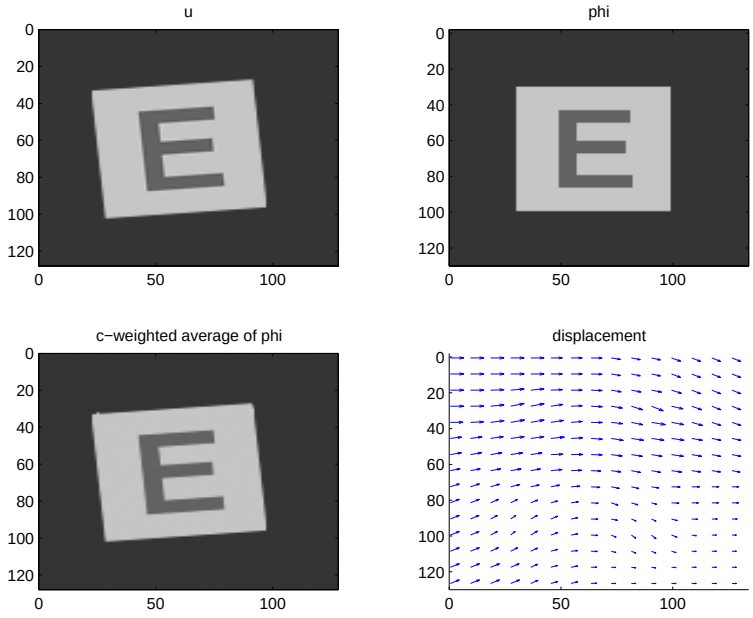


Figure 4.4: Registration of rotated and translated letter E

The pencil registration result plotted in Figure 4.5 uses the same parameters as the E example except that five by five search windows are used after the coarse solution is obtained. Recall that the initial displacement is chosen to make this

scale	iterations
2	14276
1	6560
0	10403

Table 4.2: Iterations required for registering pencil example

problem challenging by lining up the right pencil in u with the left pencil in ϕ. The displacement plot shows the location of the 46×38 image u, bordered in black, relative to the 79×98 image ϕ and draws the displacement field at a few points. The maximum displacement (r_1, r_2) is set to $(32, 16)$. Table 4.2 shows the number of iterations required.

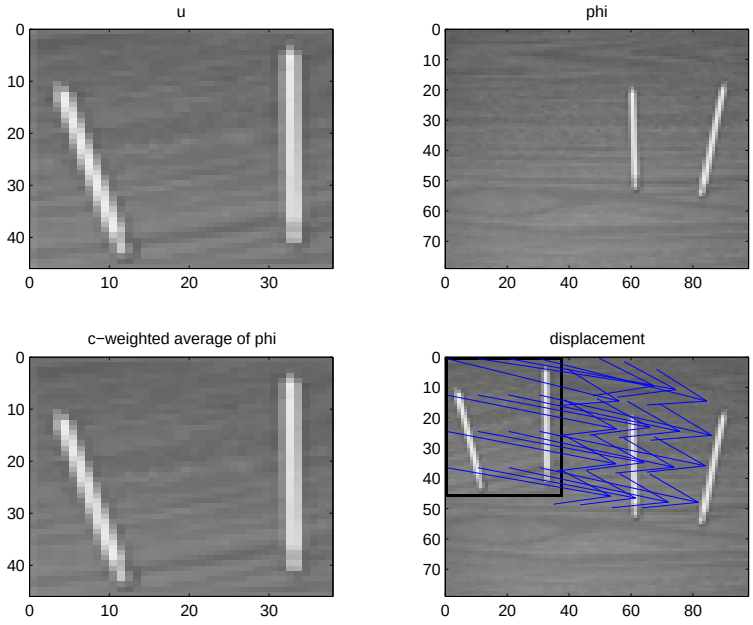


Figure 4.5: Registration of low resolution photo of two pencils

Some parameters are changed for the brain example, where u and ϕ are both 186×197 images. The maximum displacement (r_1, r_2) is set to $(10, 10)$. Since the

displacement is expected to be mostly zero, a smaller ϵ is used, namely $\epsilon = \frac{\sqrt{M}}{16}$. Adjusting α and δ to $\alpha = \frac{.0995}{s_{\phi_2}\sqrt{a}}$, $\delta = \frac{9.95 s_{\phi_2}}{\sqrt{a}}$ sped up the rate of convergence slightly for this example. Also, at the four scales computed in the multiscale approach, a three by three search window was used for the finest while five by five search windows were used for the two intermediate scales after the coarsest. The registration result is shown in Figure 4.6 and the number of iterations required are listed in Table 4.3. For this example, the ground truth displacement is known and compared to the computed displacement in Figure 4.7. The root mean square errors relative to the ground truth for the displacement components v_1 and v_2 are .3995 and .5748 respectively.

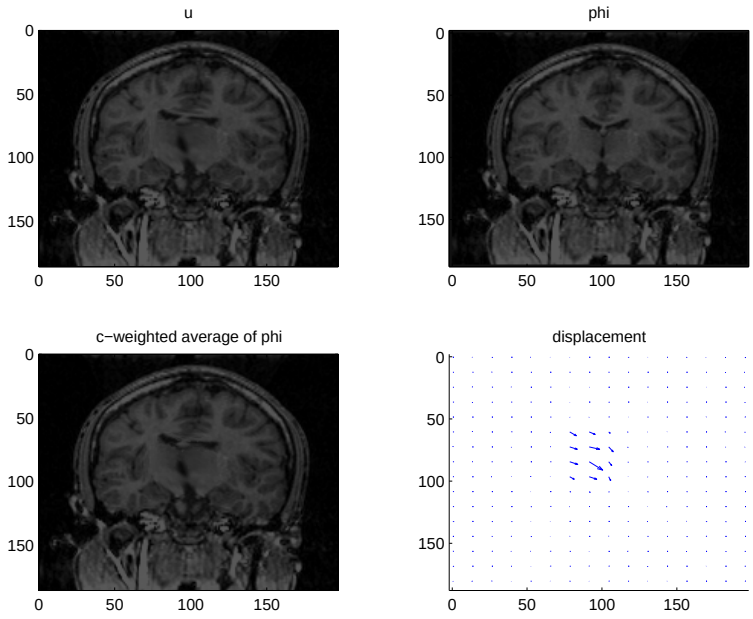


Figure 4.6: Registration of brain images

Although the convex registration model was successful on the previous examples, it can fail if the parameters are not well chosen. Three examples of what can go wrong are when the data fidelity constraints are too weak, when η is too

scale	iterations
3	3914
2	4919
1	6813
0	11303

Table 4.3: Iterations required for registering brain example

small or even when η is too large.

The pencil example will be used to illustrate the first two potential problems. The solution of the coarse, twice downsampled, problem suffices to demonstrate this. Figure 4.8 shows the c-weighted average of ϕ, the displacement, the weights corresponding to index $w = 77$ and the weights corresponding to pixel $m = 102$ that result from choosing $\epsilon = 100\sqrt{M}$ or choosing $\eta = 10^{-12}$. Since the l_∞ constraint was already weak, the c-weighted average is not a good approximation to u when ϵ is large. Moreover, the computed displacement is poor. With $\epsilon = \frac{\sqrt{M}}{2}$ but η too small, the c-weighted average accurately approximates u but the resulting displacement is not smooth or accurate. Part of the reason for these poor results is the nonlocal weights. Weights that should be concentrated on one pencil instead appear on both, which means the weighted averages of the locations in ϕ are not even close to where the weights are large. The nonlocal weights that result from poor parameter choices are also illustrated in Figure 4.8.

A large value of η actually works fine for the pencil example, but not for the brain example. Looking at the coarse, three times downsampled problem, Figure 4.9 compares the good brain image registration result for $\eta = 1$ to the poor result when $\eta = 10^{12}$. In both examples, $\epsilon = \frac{\sqrt{M}}{16}$ and the stopping condition was $\|c^{k+1} - c^k\|_\infty \leq \frac{.0001}{W}$. In the case where η is too large, the displacement is

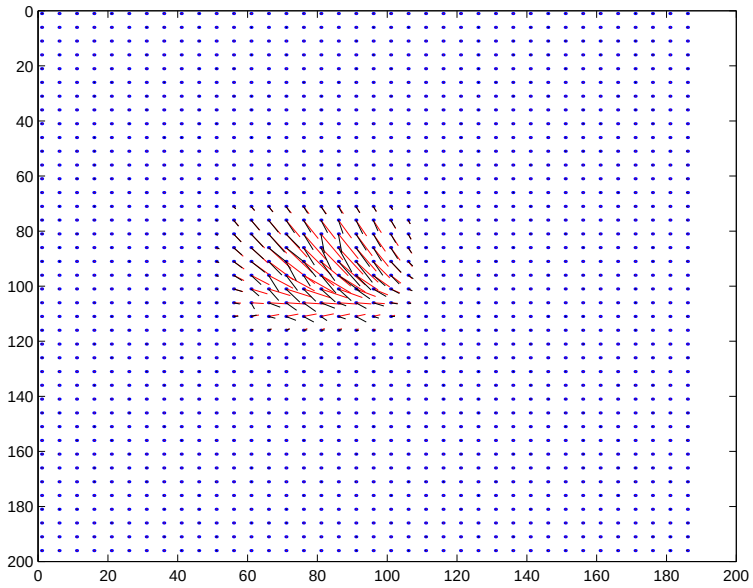


Figure 4.7: Comparison of brain registration to ground truth displacement

zero in many places where it shouldn't be. Figure 4.9 shows that for $m = 247$ corresponding to such a location, the c_m weights are more localized for the $\eta = 1$ case than for the large η case. The figure also shows that example c_w for $w = 19$ and $w = 13$ are less smooth in the large η case.

4.5 Modifications and Other Applications

4.5.1 Using Other Norms

Some image registration problems are better modeled using the l_1 norm or the TV seminorm for either the data fidelity or the regularization terms. This is the case when one expects discontinuities in the displacement field or a sparse difference between the original and registered images. If the images have slight changes in intensity, it makes sense to register the gradients of u and ϕ. Even with all these

possible changes to the model, the PDHGMp method is still applicable and the overall numerical approach is very similar.

The optical flow problem of registering successive frames in a video sequence is an example where it would make sense to use the l_1 norm for data fidelity and total variation to regularize the displacement field. The convex registration model, however, is not ideal for optical flow problems. When such problems already satisfy a small deformation assumption, the convexity of the model isn't really needed. Moreover, speed of the algorithm is especially important for video applications and the convex model is slow because of the many extra variables it must take into account. Nevertheless, it serves as a good illustration of how to substitute other norms into (4.7) and still apply PDHGMp to minimize the functional.

Consider the following TV-l_1 model for optical flow,

$$F(c) = g_C(c) + \sum_{w=1}^{W} \|\mathcal{X}_w c\|_{TV} + g_T(A_\phi c) + \eta\|A_{y_1}c - x_1\|_{TV} + \eta\|A_{y_2}c - x_2\|_{TV},$$

where g_T is the indicator function for $T = \{z : \|z - u\|_1 \leq \epsilon\}$ and the total variation seminorm is defined in Section 2.4.1.

Let g_X be the indicator function for $X = \{p : \|p\|_{E^*} \leq 1\}$ (3.3). Then the functional can be rewritten as

$$F(c) = g_C(c) + \sum_{w=1}^{W} g_X^*(D\mathcal{X}_w c) + g_T(A_\phi c) + \eta g_X^*(D(A_{y_1}c - x_1)) + \eta g_X^*(D(A_{y_2}c - x_2)).$$

Applying PDHGMp analogous to the way it was applied in Section 4.3.1, again

with scaling factors, yields the following iterations,

$$
\begin{aligned}
c^{k+1} &= \Pi_C \left(c^k - \alpha \sum_{w=1}^{W} \frac{\mathcal{X}_w^T D^T (2p_w^k - p_w^{k-1})}{s_w} - \alpha \frac{A_\phi^T (2p_\phi^k - p_\phi^{k-1})}{s_\phi} \right. \\
&\qquad \left. - \alpha \frac{A_{y_1}^T D^T (2p_{y_1}^k - p_{y_1}^{k-1})}{s_{y_1}} - \alpha \frac{A_{y_2}^T D^T (2p_{y_2}^k - p_{y_2}^{k-1})}{s_{y_2}} \right) \\
p_w^{k+1} &= s_w \Pi_X \left(\frac{(p_w^k + \frac{\delta D \mathcal{X}_w c^{k+1}}{s_w})}{s_w} \right) \qquad \text{for } w = 1, ..., W \\
p_\phi^{k+1} &= p_\phi^k + \frac{\delta A_\phi c^{k+1}}{s_\phi} - \frac{\delta}{s_\phi} \Pi_T \left((p_\phi^k + \frac{\delta A_\phi c^{k+1}}{s_\phi}) \frac{s_\phi}{\delta} \right) \\
p_{y_1}^{k+1} &= \eta s_{y_1} \Pi_X \left(\frac{(p_{y_1}^k + \frac{\delta D (A_{y_1} c^{k+1} - x_1)}{s_{y_1}})}{\eta s_{y_1}} \right) \\
p_{y_2}^{k+1} &= \eta s_{y_2} \Pi_X \left(\frac{(p_{y_2}^k + \frac{\delta D (A_{y_2} c^{k+1} - x_2)}{s_{y_2}})}{\eta s_{y_2}} \right),
\end{aligned}
$$

where $\Pi_X(p)$ is defined by (3.10). Although there isn't an explicit formula for Π_T, the orthogonal projection onto T, its computation is very similar to that for Π_C as described at the end of Section 4.3.1 and can be computed with $O(M \log M)$ complexity.

4.5.2 Different Multiscale Strategies

Although the convex registration model presented here doesn't require a multiscale numerical approach, it is impractically slow without it. Even with the coarse to fine approach, it took 20 to 30 minutes for the E and pencil examples and several hours for the brain example. Most of this time was needed for computing the solutions at the finest scales. However, after solving the coarse problem, a small deformation assumption is satisfied for all problems at finer resolutions. Therefore, it might make sense to solve the convex model only for the coarsest problem in the multiscale approach, switching perhaps to a more

efficient linearized version of something like (4.1) for the finer scales.

4.5.3 More Implicit Numerical Methods

The special structure of the matrices A_ϕ, A_{y_1}, A_{y_2} and $D\mathcal{X}_w$ suggests that a more implicit algorithm than PDHGMp might work well. Since $A_\phi A_\phi^T$, $A_{y_1}A_{y_1}^T$ and $A_{y_2}A_{y_2}^T$ are diagonal, terms like $I + A_\phi^T A_\phi$, $I + A_{y_1}^T A_{y_1}$ and $I + A_{y_2}^T A_{y_2}$ are easy to invert using the Sherman Morrison Woodbury formula. A term like $I + \sum_w \mathcal{X}_w^T D^T D \mathcal{X}_w$ is also easy to deal with because with the proper indexing it is a block diagonal matrix with I minus the discrete Laplacian as each block. So, with the addition of some extra variables, the application of split Bregman [GO09] yields simple iterations. The equivalent application of ADMM (3.26 to the Lagrangian

$$\begin{aligned}
L = {} & g_C(c) \\
& + \sum_{w=1}^{W} \left(g_B^*(z_w) + \langle p_w, D\mathcal{X}_w u_1 - z_w \rangle \right) + \langle r_1, c - u_1 \rangle \\
& + g_{T_2}(z_{\phi_2}) + \langle p_{\phi_2}, A_\phi u_2 - z_{\phi_2} \rangle + \langle r_2, c - u_2 \rangle \\
& + g_{T_\infty}(z_{\phi_\infty}) + \langle p_{\phi_\infty}, A_\phi u_3 - z_{\phi_\infty} \rangle + \langle r_3, c - u_3 \rangle \\
& + R_1(z_{y_1}) + \langle p_{y_1}, A_{y_1} u_4 - z_{y_1} \rangle + \langle r_4, c - u_4 \rangle \\
& + R_2(z_{y_2}) + \langle p_{y_2}, A_{y_2} u_5 - z_{y_2} \rangle + \langle r_5, c - u_5 \rangle
\end{aligned}$$

was attempted but found to be slightly less efficient for the examples tested. With the introduction of scaling parameters similar to those used for PDHGMp, ADMM required similar numbers of iterations to meet the same stopping criteria. However, the more implicit ADMM iterations were more time consuming and memory intensive. It may still be possible to improve the performance with better parameter choices or by working with a different Lagrangian formulation

that involves fewer variables but requires solving slightly more complicated linear systems for some of the subproblems.

4.5.4 Dimension Reduction

A possible idea for speeding up the method without resorting to a multiscale approach is to try to approximate c by a linear transformation of a lower dimensional vector s. The motivation for this kind of dimension reduction is that the $c_w \in \mathbb{R}^M$ represent smooth images and can therefore be well approximated by, for example, the low frequency terms in its representation via the discrete cosine transform (DCT). Putting these $l < M$ low frequency DCT basis vectors in the columns of $\Psi \in \mathbb{R}^{M \times l}$, we can try to represent $c_w = \Psi s_w$ for $w = 1, ..., W$. A difficulty with this approach is finding feasible constraints on s so that $c \in C$. The situation can be somewhat simplified by using the above constraint relaxation idea, but we still must have $\Psi s_w \geq 0$ and $\Psi \sum_w s_w = 1$. Assuming $\Psi^T \Psi = I$, the overall constraints on s could be written

$$\sum_{w=1}^{W} s_w = \Psi^T 1 \qquad , \qquad \Psi s_w \geq 0 \qquad \forall w.$$

If this approach is to succeed, it will likely be necessary to redesign Ψ so that the constraints on s are feasible and also computable without reintroducing the larger vector c that we are trying to approximate in the first place.

4.5.5 Constraint Relaxation

It's possible to slightly speed up the iterations for PDHGMp applied to (4.8) by splitting up the constraint $c \in C$ into two separate constraints. Recall that this normalization constraint enforces that for each pixel $m \in \mathbb{R}^m$, the corresponding vector of weights $c_m \in \mathbb{R}^W$ has to be nonnegative and sum to one. Consider

adding a new variable s constrained to equal c such that s is constrained to be nonnegative and each vector c_m is constrained to sum to one. Numerically this can be handled by introducing indicator functions for these constraints into the model and applying the split inexact Uzawa method from [ZBO09] to the Lagrangian

$$g_{\{s\geq 0\}}(s) + g_{\{\sum_w c_{m,w}=1\}}(c) + J(z) + \langle p, Ac - z\rangle + \langle \lambda, c - s\rangle,$$

where p and λ are Lagrange multipliers for the $Ac = z$ and $s = c$ constraints. Compared to the PDHGMp implementation, the c update is replaced by

$$\begin{aligned} c_m^{k+1} &= \frac{1}{W} + (c^k - \alpha A^T(2p^k - p^{k-1}) - \alpha\lambda^k)_m - \\ &\quad \text{mean}((c^k - \alpha A^T(2p^k - p^{k-1}) - \alpha\lambda^k)_m) \text{ for } m = 1, ..., M, \\ s^{k+1} &= \max(c^{k+1} + \frac{\lambda^k}{\delta}, 0), \end{aligned}$$

the p updates are identical, and there is an additional update for the multiplier λ,

$$\lambda^{k+1} = \lambda^k + \delta(c^{k+1} - s^{k+1}).$$

The projection onto C which could be computed with complexity $O(MW\log(W))$ has now been replaced with two simpler projections that have complexity $O(MW)$. This does indeed speed up each iteration, but more iterations are also required, especially when M is large. Overall the method tends not to be any faster unless W is large or M is small.

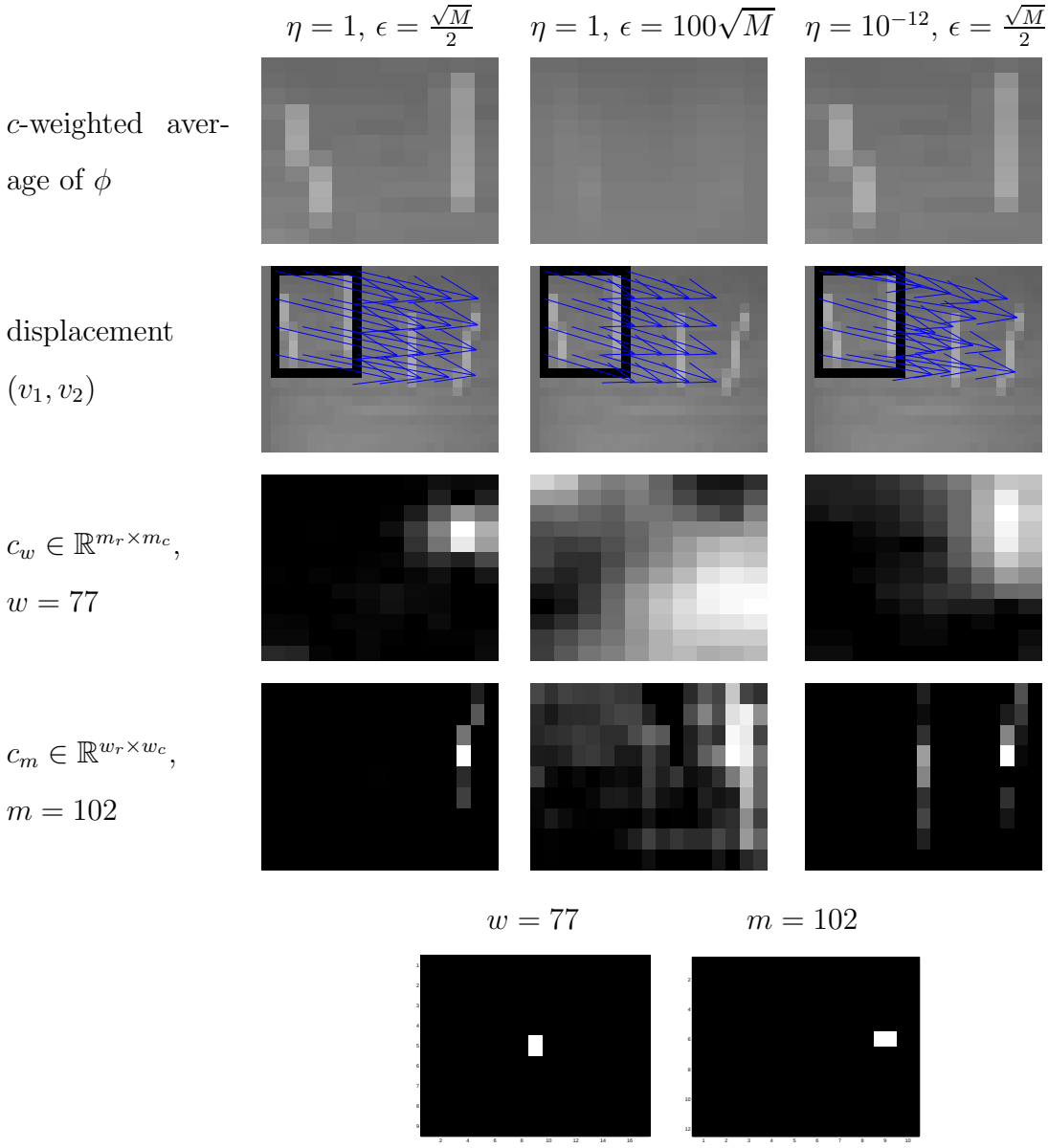


Figure 4.8: Comparison of coarse pencil registration results

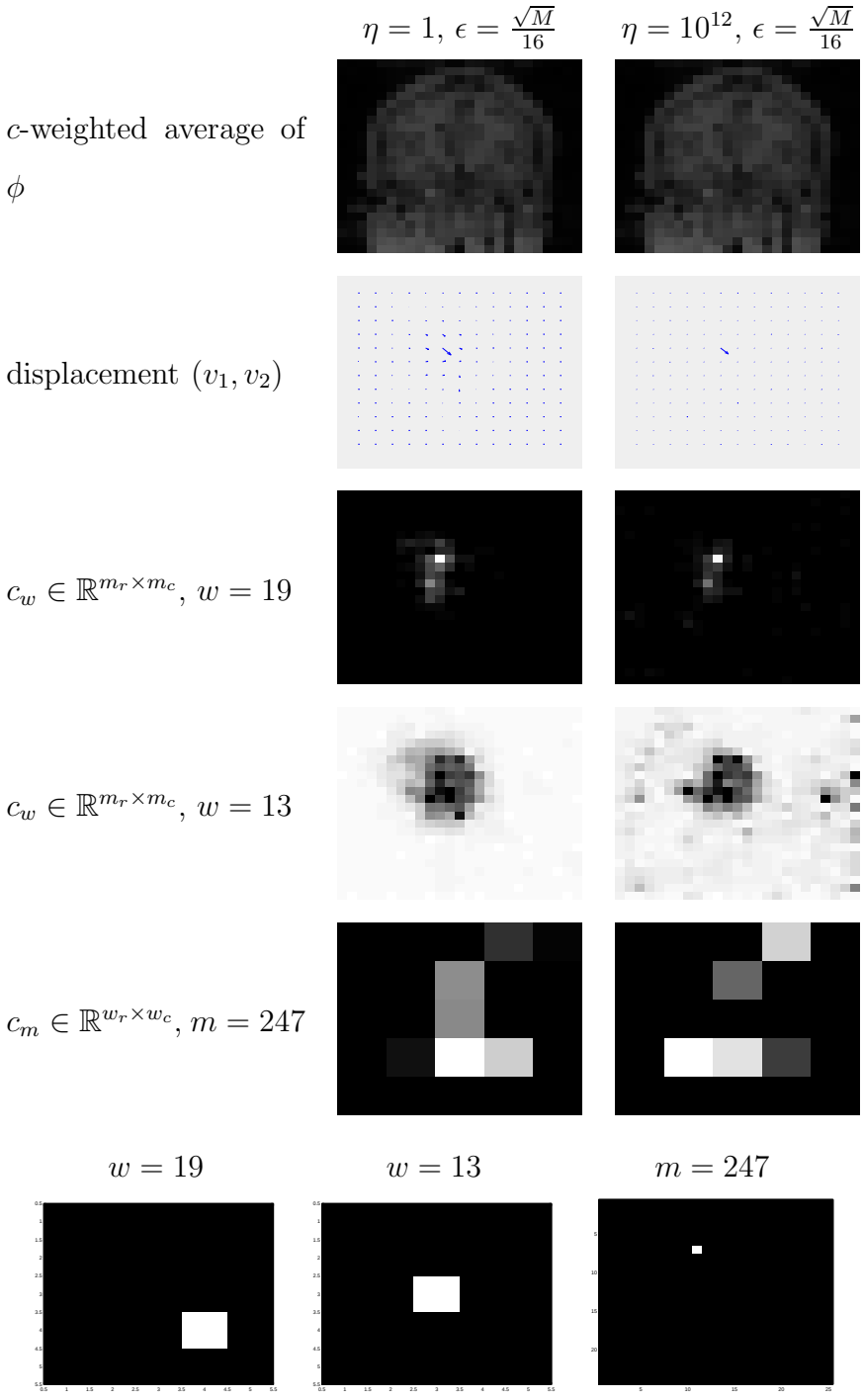


Figure 4.9: Comparison of coarse brain image registration results

CHAPTER 5

A Convex Model for Patch-Based Nonlocal Image Inpainting

5.1 Introduction

In this chapter, a convex variational model for nonlocal image inpainting is proposed. It uses patches from anywhere in the known part of the image to fill in a large unknown area. There are many existing convex inpainting models, but they tend to be based on propagating local information into the unknown region and therefore aren't well suited for filling in areas far from the boundary. For example, total variation inpainting [CS05] works well for piecewise constant images and when missing pixels are close to known pixels. It is good for geometry inpainting and interpolation but not for texture inpainting.

Greedy approaches for exemplar-based texture inpainting have been successfully considered by many authors. The idea is closely related to the texture synthesis technique of [EL99] that sweeps through the unknown pixels, greedily setting each to be the value from the center of the image patch that best agrees with its known neighboring pixels.

Previous variational methods for texture inpainting have also been proposed. A variational model proposed in [DSC03] and extended in [ALM08] is based on a correspondence map Γ, which maps from the unknown region to the known

region such that the value u at the location x is given by $u(x) = u(\Gamma(x))$. The functionals to be minimized essentially require that $u(x-y)$ be close to $u(\Gamma(x)-y)$ for y in a neighborhood of 0, and that Γ should locally behave like a translation operator. A more easily computable variational approach in [ACS09] minimizes a nonlocal means type energy but with dynamic weights that are determined by also minimizing an entropy term. It produces the same kinds of updates for the dynamic weights as the nonlocal total variation wavelet inpainting method in [ZC09]. These variational approaches are all based on nonconvex models.

The approach proposed here is inspired by the method of Arias, Caselles and Sapiro in [ACS09]. We first consider a small set of unknown patches that cover the inpainting region. Each unknown patch will be a c-weighted average of known patches with the weights c to be determined by minimizing a convex functional. All patches are of uniform size. To yield a good solution, the weights c should be sparse, since each unknown patch should be a weighted average only of very similar patches. For many examples, the ideal situation would be for the weights to be binary where each unknown patch would exactly equal a known patch. Unknown pixels are defined to be a weighted average of contributing pixels from overlapping unknown patches with these weights fixed in advance and emphasizing more heavily pixels near patch centers. The proposed functional consists of two terms. The first term penalizes at each pixel in or near the inpainting region the sum of the squares of the differences between the values of the contributing pixels and the value at the current pixel. This encourages the unknown patches to agree with each other where they overlap and to agree with any known data they overlap. The second term regularizes the weights by treating them like correspondence maps. The weights for a single unknown patch correspond to the locations of the centers of known patches. Weights for a neighboring unknown patch shifted by v should more likely than not correspond

to the same previous locations also shifted by v. The second term of the functional enforces this by penalizing the l_1 norm of the differences of these weights, wherever these differences are defined.

Since the proposed model is convex, it can be solved using the PDHG algorithm and its related variants discussed in Chapter 3. However, the global minimum of this model is not always a great solution. The recovered image tends to be somewhat blurry and averaged out, especially away from the boundary. The method can still work reasonably well for simple examples with repetitive structure. An example of this is given in Section 5.3.2. The blurriness occurs when the weights don't converge to a sparse enough solution. The unknown patches can therefore end up being averages of too many known patches. This causes a loss of contrast in the unknown patches, which in turn can actually help them agree with each other where they overlap. Moreover, having many nonzero weights can still be consistent with the correspondence constraint. Although there would almost surely be disagreement near the boundary for non-sparse weights, this isn't enough by itself to enforce sparsity.

It's difficult to encourage c to be sparse while maintaining convexity of the model. The constraint on c helps by requiring that the weights in each weighted average be nonnegative and sum to one. Another strategy is to use the l_1 norm instead of the l_2 norm for the data fidelity term. This would determine unknown pixels by taking a weighted median of the corresponding pixels from overlapping patches. This modification is discussed in Section 5.4.1. It still doesn't always produce the desired sparsity of the weights.

To further encourage sparsity of c, the convex model can be modified by adding a nonconvex term of the form $\gamma(\langle c, 1\rangle - \|c\|^2)$. This is analogous to how a double well potential is used to enforce the binary constraint in phase field

approaches for image segmentation, except here it suffices to use a quadratic function because c is already constrained to lie between zero and one. This modification can lead to much better solutions with binary weights as demonstrated in Section 5.4.2. Unfortunately, the resulting model is no longer convex, and the numerical scheme is also not guaranteed to converge to a global minimum. We still use a modified version of PDHGMp (3.35) to produce the examples in Section 5.4.2, but its convergence is no longer guaranteed.

5.2 Notation and Formulation of Model

The formulation of the model is notationally heavy despite being based on simple ideas. Figure 5.1 shows a picture of the setup, and the key notation is defined in the list in Section 5.2.1.

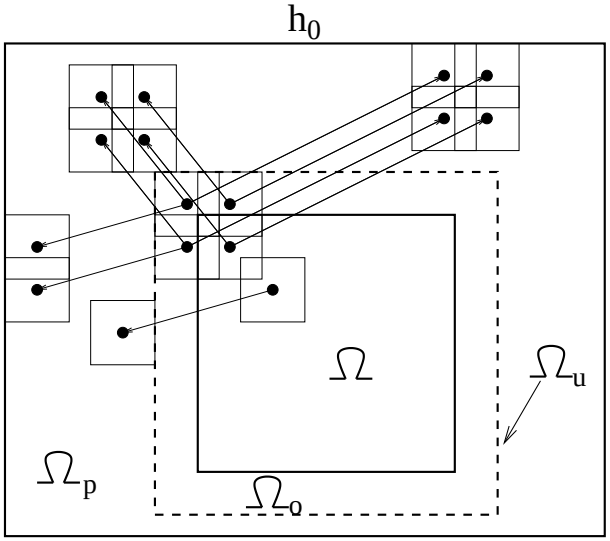


Figure 5.1: Regions Defined for Nonlocal Inpainting Model

5.2.1 Notation

$h \in \mathbb{R}^{m_r \times m_c}$	Original image
Ω	Inpainting region: Set of pixels (i, j) such that $h(i, j)$ is unknown
p_s	Patch size (assume $p_s = 6n + 3$ for simplicity)
Ω_u	Region of unknown patches: Set of pixels (i, j) covered by unknown patches. This should strictly contain Ω.
v	Index for pixels in Ω_u, $v = 1, ..., \lvert\Omega_u\rvert$
Ω_{up}	Subset of Ω_u consisting of a grid of pixels spaced apart by $\frac{2p_s}{3}$, corresponding to the unknown patches that will be solved for
Ω_o	Overlap region: $\Omega_o = \Omega_u \bigcap \Omega^c$
Ω_p	Region of known patches: Set of pixels for which corresponding patches are contained in Ω^c
$P \in \mathbb{R}^{p_s^2 \times \lvert\Omega_p\rvert}$	Matrix of vectorized known patches. $P(q, p)$ is the q^{th} pixel in the p^{th} patch, $q = 1, ..., p_s^2$, $p = 1, ..., \lvert\Omega_p\rvert$
$u \in \mathbb{R}^{\lvert\Omega_u\rvert}$	Value at pixels in Ω_u constrained so $u(v) = h(v)$ for $v \in \Omega_o$
S	Set of valid u: $\{u : u(v) = h(v) \text{ for } v \in \Omega_o\}$
$c \in \mathbb{R}^{\lvert\Omega_p\rvert \times \lvert\Omega_{up}\rvert}$	Weights for representing Ω_{up} patches as weighted averages of Ω_p patches. Must constrain $c(p, m) \geq 0$ and $\sum_p c(p, m) = 1$ $\forall m$, $m = 1, ..., \lvert\Omega_{up}\rvert$
C	Set of valid c: $\{c : c(p, m) \geq 0 \text{ and } \sum_p c(p, m) = 1 \quad \forall m\}$
$Pc \in \mathbb{R}^{p_s^2 \times \lvert\Omega_{up}\rvert}$	Matrix product of P and c is matrix of vectorized Ω_{up} patches
$\beta(q)$	Vectorized 2D Gaussian weights (standard deviation $\frac{p_s}{3}$) defined on a single patch,
$\beta_v(q)$	Normalized weights $\beta_v(q) = \frac{\beta(q)}{\sum_{Q_v} \beta(q)}$ where $Q_v = \{q :$ there exists a Ω_{up} patch whose q^{th} pixel overlaps $v\}$

5.2.2 Definition of Functional

The proposed functional will initially be defined in terms of c and u. Later, the expression for u in terms of c will be substituted in. The constraints on c and u will be handled by introducing indicator functions $g_C(c)$ and $g_S(u)$ for the sets C and S.

$$g_C(c) = \begin{cases} 0 & \text{if } c \in C \\ \infty & \text{otherwise} \end{cases} \qquad g_S(u) = \begin{cases} 0 & \text{if } u \in S \\ \infty & \text{otherwise} \end{cases}$$

The first term of the functional can be written

$$\sum_{v=1}^{|\Omega_u|} \sum_{\text{contributing}(q,m)} \left(\beta_v(q)((Pc)(q,m) - u(v))\right)^2,$$

where the contributing (q, m) indices are those for which the q^{th} pixel in the m^{th} Ω_{up} patch overlaps pixel v. This can be more conveniently rewritten as

$$||A(c) - B(u)||_F^2,$$

where $\|\cdot\|_F$ is the Frobenius norm and $A : \mathbb{R}^{|\Omega_p| \times |\Omega_{up}|} \to \mathbb{R}^{p_s^2 \times |\Omega_u|}$ and $B : \mathbb{R}^{|\Omega_u|} \to \mathbb{R}^{p_s^2 \times |\Omega_u|}$ are linear operators defined as follows.

$$A(c)(q,v) = \begin{cases} \beta_v(q)(Pc)(q,m) & \text{if there exists } m \text{ such that pixel } q \\ & \text{of the } \Omega_{up} \text{ patch at } m \text{ overlaps } v \\ 0 & \text{otherwise} \end{cases}$$

$$B(u)(q,v) = \begin{cases} \beta_v(q)u(v) & \text{if there exists } m \text{ such that pixel } q \\ & \text{of the } \Omega_{up} \text{ patch at } m \text{ overlaps } v \\ 0 & \text{otherwise} \end{cases}$$

Note that u is only compared to pixels that come from weighted averages of known patches. This is a potential weakness of this choice of data fidelity term. It would be better to directly compare u to the information in the known patches

in this weighted average, but we don't because to do so would be much more computationally intensive.

The correspondence term of the functional will be defined as

$$\sum_{m=1}^{|\Omega_{up}|} \sum_{\tilde{m} \sim m} \sum_{p=1}^{|\Omega p|} \begin{cases} |c(p,m) - c(\tilde{p}(p,\tilde{m},m),\tilde{m})| & \text{if defined} \\ 0 & \text{otherwise,} \end{cases}$$

where $\tilde{p}(p,\tilde{m},m)$ denotes the index for the Ω_p patch shifted from p by the same amount $\tilde{m}$ is shifted from m, defined when contained in Ω_p. These differences can be more conveniently written in terms of a linear operator $D : \mathbb{R}^{|\Omega_p| \times |\Omega_{up}|} \to \mathbb{R}^e$, with e the total number of differences taken. With this notation the correspondence term can be rewritten as $\|D(c)\|_1$.

The proposed functional is then

$$G(c,u) = g_C(c) + g_S(u) + \frac{\mu}{2}\|A(c) - B(u)\|_F^2 + \|D(c)\|_1, \tag{5.1}$$

which is to be minimized with respect to u and c. Note, however, that u can be easily solved for in terms of c.

$$u(v) = \begin{cases} h(v) & v \in \Omega_o \\ ((B^*B)^{-1}B^*A(c))(v) & v \in \Omega, \end{cases}$$

where B^* denotes the adjoint of B. This formula for u can be thought of as taking a weighted average at each unknown pixel of the contributing pixels from overlapping patches. Pixels closer to the center of the patches are more heavily weighted according to the Gaussian weights β. This weighted averaging update for u is very similar to the one used in [ACS09]. When plugging the formula for u back into $\|A(c) - B(u)\|_F^2$, it makes sense to break the term into two parts, one

defined on Ω_o and the other defined on Ω. To that end, define

$$\mathcal{X}_{\Omega_o}(q,v) = \begin{cases} 1 & v \in \Omega_o \\ 0 & \text{otherwise} \end{cases} \qquad \mathcal{X}_{\Omega}(q,v) = \begin{cases} 1 & v \in \Omega \\ 0 & \text{otherwise} \end{cases}.$$

Also define

$$h_0(v) = \begin{cases} h(v) & v \in \Omega_o \\ 0 & \text{otherwise} \end{cases}.$$

Plugging the expression for u into G yields

$$g_C(c) + \frac{\mu}{2}\|\mathcal{X}_{\Omega_o} \cdot A(c) - \mathcal{X}_{\Omega_o} \cdot B(h_0)\|_F^2 + \frac{\mu}{2}\|\mathcal{X}_{\Omega} \cdot (I - B(B^*B)^{-1}B^*)A(c)\|_F^2 + \|D(c)\|_1,$$

where $\cdot$ denotes componentwise multiplication of matrices. Let

$$f = \mathcal{X}_{\Omega_o} \cdot B(h_0)$$

and define linear operators A_{Ω_o} and A_{Ω} such that

$$A_{\Omega_o}(c) = \mathcal{X}_{\Omega_o} \cdot A(c)$$

and

$$A_{\Omega}(c) = \mathcal{X}_{\Omega} \cdot (I - B(B^*B)^{-1}B^*)A(c).$$

Now we can define a convex functional in terms of c,

$$F(c) = g_C(c) + \frac{\mu_{\Omega_o}}{2}\|A_{\Omega_o}(c) - f\|_F^2 + \frac{\mu_{\Omega}}{2}\|A_{\Omega}(c)\|_F^2 + \|D(c)\|_1, \qquad (5.2)$$

and attempt to solve the inpainting problem by finding a minimizer.

5.3 Numerical Approach

Variants of the PDHG method (3.18) are well suited for minimizing F (5.2). In this section, we demonstrate how to apply the algorithm and also present several numerical examples that show some of the strengths and weaknesses of the model.

5.3.1 Application of PDHGMp

To minimize F, we use the PDHGMp variant (3.35) of the PDHG method. Let

$$H(c) = g_C(c),$$

$$\tilde{A} = \begin{bmatrix} A_{\Omega_o} \\ A_\Omega \\ D \end{bmatrix}$$

and

$$J(\tilde{A}(c)) = J_{\Omega_o}(A_{\Omega_o}(c)) + J_\Omega(A_\Omega(c)) + J_D(D(c)),$$

where

$$J_{\Omega_o}(z_{\Omega_o}) = \frac{\mu_{\Omega_o}}{2}\|z_{\Omega_o} - f\|_F^2,$$
$$J_\Omega(z_\Omega) = \frac{\mu_\Omega}{2}\|z_\Omega\|_F^2$$

and

$$J_D(z_D) = \|z_D\|_1.$$

With the addition of dual variables p_{Ω_o}, p_Ω, p_D, time step parameters α, δ and optional scaling parameters s_{Ω_o}, s_Ω, s_D as discussed in Section 4.3.1, the PDHGMp iterations are given by

$$\begin{aligned}
c^{k+1} &= \arg\min_c g_C(c) + \frac{1}{2\alpha}\|c - \left(c^k - \alpha\frac{A_{\Omega_o}^*(2p_{\Omega_o}^k - p_{\Omega_o}^{k-1})}{s_{\Omega_o}} - \alpha\frac{A_\Omega^*(2p_\Omega^k - p_\Omega^{k-1})}{s_\Omega}\right. \\
&\quad \left. -\alpha\frac{D^*(2p_D^k - p_D^{k-1})}{s_D}\right)\|_F^2 \\
p_{\Omega_o}^{k+1} &= \arg\min_{p_{\Omega_o}} J_{\Omega_o}^*(\frac{p_{\Omega_o}}{s_{\Omega_o}}) + \frac{1}{2\delta}\|p_{\Omega_o} - (p_{\Omega_o}^k + \frac{\delta A_{\Omega_o}(c^{k+1})}{s_{\Omega_o}})\|_F^2 \\
p_\Omega^{k+1} &= \arg\min_{p_\Omega} J_\Omega^*(\frac{p_\Omega}{s_\Omega}) + \frac{1}{2\delta}\|p_\Omega - (p_\Omega^k + \frac{\delta A_\Omega(c^{k+1})}{s_\Omega})\|_F^2 \\
p_D^{k+1} &= \arg\min_{p_D} J_D^*(\frac{p_D}{s_D}) + \frac{1}{2\delta}\|p_D - (p_D^k + \frac{\delta A_D(c^{k+1})}{s_D})\|_F^2,
\end{aligned}$$

where the initialization is arbitrary. Each of the above minimizers can be explicitly solved by the following formulas,

$$c^{k+1} = \Pi_C \left(c^k - \alpha \frac{A^*_{\Omega_o}(2p^k_{\Omega_o} - p^{k-1}_{\Omega_o})}{s_{\Omega_o}} - \alpha \frac{A^*_{\Omega}(2p^k_{\Omega} - p^{k-1}_{\Omega})}{s_{\Omega}} - \alpha \frac{D^*(2p^k_D - p^{k-1}_D)}{s_D} \right) \tag{5.3a}$$

$$p^{k+1}_{\Omega_o} = \frac{p^k_{\Omega_o} + \frac{\delta}{s_{\Omega_o}}(A_{\Omega_o}(c^{k+1}) - f)}{\frac{\delta}{\mu_{\Omega_o} s^2_{\Omega_o}} + 1} \tag{5.3b}$$

$$p^{k+1}_{\Omega} = \frac{p^k_{\Omega} + \frac{\delta}{s_{\Omega}} A_{\Omega}(c^{k+1})}{\frac{\delta}{\mu_{\Omega} s^2_{\Omega}} + 1} \tag{5.3c}$$

$$p^{k+1}_D = \Pi_{\{z:\|z\|_\infty \le s_D\}} \left(p^k_D + \frac{\delta}{s_D} D(c^{k+1}) \right) \tag{5.3d}$$

where Π_C and $\Pi_{\{z:\|z\|_\infty \le s_D\}}$ denote orthogonal projection onto C and $\{z : \|z\|_\infty \le s_D\}$ respectively. The projection $\Pi_C(c)$ amounts to projecting each column of c onto the positive face of the l_1 unit ball. This ensures the weights are nonnegative and normalized. The same projection appears in Section 4.3.1, but there it is applied to the rows of a matrix.

5.3.2 Numerical Results

The convex inpainting model works best for simple images with repeating structure and we show its successful application to the problem of inpainting a missing portion of a brick wall in Figures 5.2 and 5.3. These examples also demonstrate the effect of the correspondence term $\|D(c)\|_1$, which encourages information in the recovered image to have similar spatial correspondence as information in the known part of the image. In the extreme case where the weights are binary and the correspondence term equals zero, the recovered data would simply be a copy of a contiguous block of known image. Figure 5.2 shows the inpainting result without the correspondence term. The parameters μ_{Ω_o} and μ_{Ω} were both set

equal to one. The scaling parameters, which affect the efficiency of the numerical scheme but don't change the model, were chosen to be $s_{\Omega_o} = 100000$ and $s_{\Omega} = 10000$. Figure 5.3 shows the result with the correspondence term included. For this example, $1000\|D(c)\|_1$ was added to the functional and $s_D = 100$. As can be seen in the figures, the addition of the correspondence term makes it possible to better reproduce the repeating structure of the image even when far from the boundary.

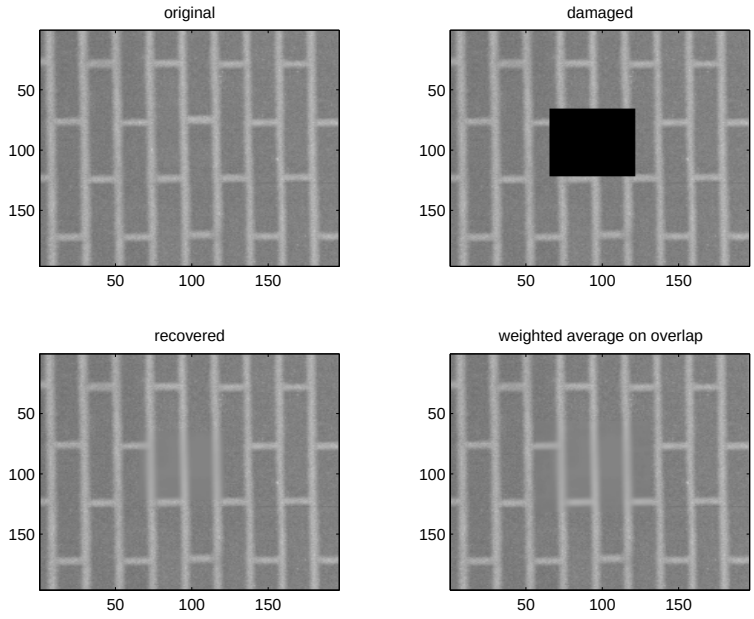


Figure 5.2: Inpainting brick wall using 15×15 patches but without including the correspondence term

For more complicated images like the picture of grass in Figure 5.4, the addition of the correspondence term is not always able to encourage recovery of more detail in the inpainting region. In this example, with 15 by 15 patches, it's difficult to find known patches that agree well with the boundary information. When that happens, weights minimizing the convex functional tend to be

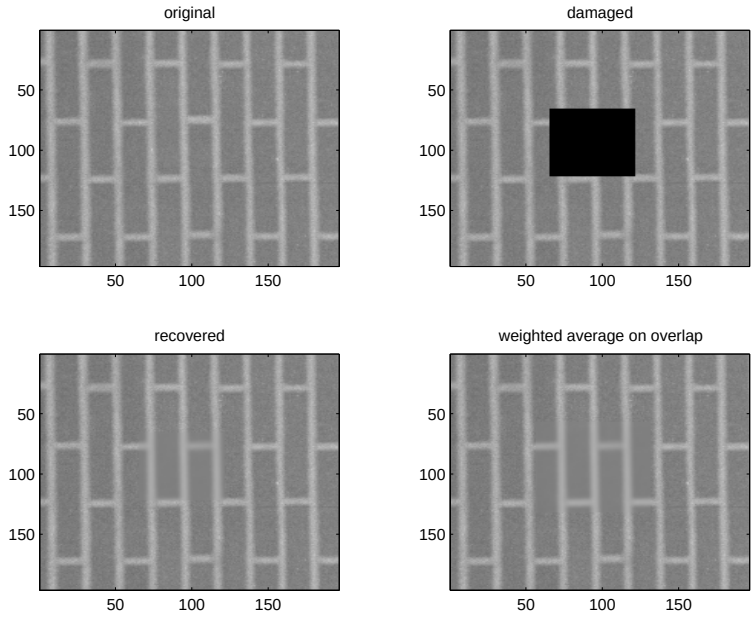


Figure 5.3: Inpainting brick wall using 15×15 patches and including the correspondence term

less sparse. That's because when the unknown patches end up being averages of many known patches, they become more nearly constant and therefore agree well with patches they overlap. Figure 5.4 shows an example of such an unsatisfactory over-averaged inpainting result. Modifications to the functional that address this drawback are discussed in the next section.

5.4 Modifications to Functional

Some modifications to F (5.2) intended to improve the sparsity of the weights are discussed in this section. Using the l_1 norm instead of the l_2 norm for the data fidelity term leads to slightly sparser weights, but the results are not significantly different. On the other hand, we show that adding a nonconvex term to encourage

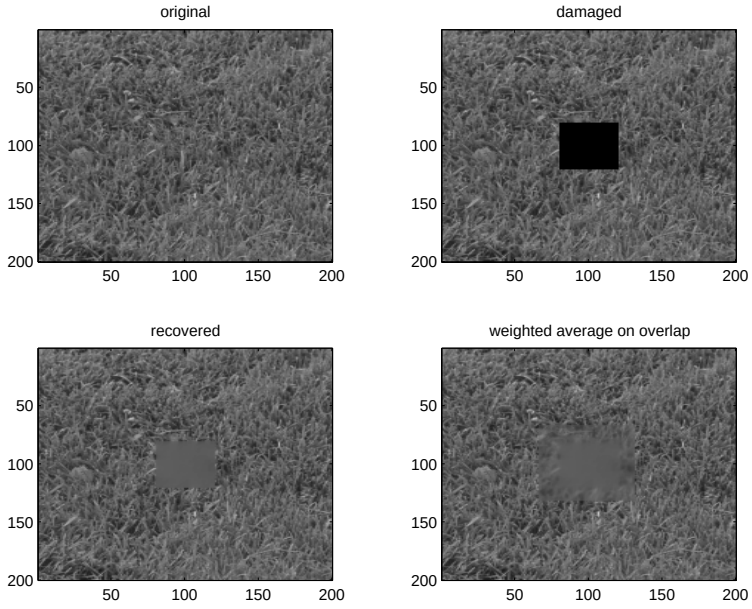


Figure 5.4: Inpainting grass using 15×15 patches and including the correspondence term

binary weights can lead to good quality sparse solutions. Unfortunately, the model in that case is no longer convex, the numerical approach becomes somewhat ad-hoc and the results can be sensitive to parameter choices. Even so, the superior results such an approach can yield merits a brief discussion.

5.4.1 Using l_1 Data Fidelity

If the l_1 norm is used instead of l_2 for the data fidelity term, then the resulting problem is

$$\min_{u,c} g_C(c) + g_S(u) + \frac{\mu}{2}\|A(c) - B(u)\|_1 + \|D(c)\|_1, \tag{5.4}$$

It's still possible to solve for u as a function of c. At each unknown pixel v, $u(v)$ is given by a weighted median of the contributing pixels from overlapping patches, which appear as the nonzero entries of the v^{th} column of $A(c)$. The weights

in this weighted median depend on the Gaussian weights β and the number of contributing pixels. However, the resulting u depends nonlinearly on c, and so we can't use the same approach of substituting this formula back into the functional.

We could instead directly apply the PDHGMp algorithm, thinking of $\begin{bmatrix} c \\ u \end{bmatrix}$ as a single variable y and letting $H(y) = g_C(c) + g_S(u)$, $\tilde{A} = \begin{bmatrix} A & -B \end{bmatrix}$ and $J(\tilde{A}(y)) = \frac{\mu}{2}\|A(c) - B(u)\|_1 + \|D(c)\|_1$. Unfortunately, the stability restriction that $\alpha\delta < \frac{1}{\|\tilde{A}^*\tilde{A}\|}$ is too severe in this case because A and B can no longer be scaled independently of each other.

An alternating version that works better in practice but is not theoretically justified is to apply PDHGMp to (5.4) as if u were fixed, and then directly update u every few iterations by computing the appropriate weighted median. In practice this did lead to slightly sparser weights for the examples tested, but the results compared to the l_2 version did not differ significantly in visual quality.

5.4.2 Adding Nonconvex Term to Encourage Binary Weights

Motivated by the phase field approach for segmentation that enforces a binary constraint by introducing a double well potential, we consider a similar strategy for making c sparser or even binary. The double well potential strategy would be to add a term of the form

$$\gamma \sum_{p,m} c_{p,m}^2 (1 - c_{p,m})^2$$

to (5.2). Since the normalization constraint $c \in C$ already forces $0 \leq c_{p,m} \leq 1$, we instead choose to add the quadratic function

$$\gamma \sum_{p,m} c_{p,m}(1 - c_{p,m}),$$

which can be rewritten as

$$\gamma\langle c, 1\rangle - \gamma\|c\|_F^2.$$

The resulting nonconvex functional is defined by

$$F_{nc}(c) = g_C(c) + \gamma\langle c, 1\rangle - \gamma\|c\|_F^2 + \frac{\mu_{\Omega_o}}{2}\|A_{\Omega_o}(c) - f\|_F^2 + \frac{\mu_\Omega}{2}\|A_\Omega(c)\|_F^2 + \|D(c)\|_1. \tag{5.5}$$

We use the same numerical approach as in Section 5.3 after first redefining

$$H(c) = g_C(c) + \gamma\langle c, 1\rangle - \gamma\|c\|_F^2.$$

The PDHGMp minimization steps in (5.3) remain the same except for the c^{k+1} update, which becomes

$$c^{k+1} = \arg\min_c g_C(c) + \gamma\langle c, 1\rangle - \gamma\|c\|_F^2 + \frac{1}{2\alpha}\|c - \left(c^k - \alpha\frac{A_{\Omega_o}^*(2p_{\Omega_o}^k - p_{\Omega_o}^{k-1})}{s_{\Omega_o}}\right. \tag{5.6}$$

$$\left. -\alpha\frac{A_\Omega^*(2p_\Omega^k - p_\Omega^{k-1})}{s_\Omega} - \alpha\frac{D^*(2p_D^k - p_D^{k-1})}{s_D}\right)\|_F^2.$$

Let $0 \leq \gamma < \frac{1}{2\alpha}$. This ensures that the objective functional for the c^{k+1} update remains convex. Naturally, if $\gamma = 0$ then the update is unchanged from before. Altogether the PDHGMp iterations are given by

$$\begin{aligned}
c^{k+1} &= \Pi_C\left(\left(\frac{1}{1-2\alpha\gamma}\right)\left(c^k - \alpha\frac{A_{\Omega_o}^*(2p_{\Omega_o}^k - p_{\Omega_o}^{k-1})}{s_{\Omega_o}} - \alpha\frac{A_\Omega^*(2p_\Omega^k - p_\Omega^{k-1})}{s_\Omega}\right.\right.\\
&\qquad \left.\left. -\alpha\frac{D^*(2p_D^k - p_D^{k-1})}{s_D} - \gamma\alpha\right)\right)\\
p_{\Omega_o}^{k+1} &= \frac{p_{\Omega_o}^k + \frac{\delta}{s_{\Omega_o}}(A_{\Omega_o}(c^{k+1}) - f)}{\frac{\delta}{\mu_{\Omega_o}s_{\Omega_o}^2} + 1}\\
p_\Omega^{k+1} &= \frac{p_\Omega^k + \frac{\delta}{s_\Omega}A_\Omega(c^{k+1})}{\frac{\delta}{\mu_\Omega s_\Omega^2} + 1}\\
p_D^{k+1} &= \Pi_{\{z:\|z\|_\infty \leq s_D\}}\left(p_D^k + \frac{\delta}{s_D}D(c^{k+1})\right)
\end{aligned}$$

Note that the convergence theory for PDHGMp based on Theorem 3.4.2 no longer applies because $F_{nc}(c)$ is not convex. In practice, the method does not find global minimizers of F_{nc}, but it does produce good solutions with binary weights as demonstrated in Section 5.4.2.1.

5.4.2.1 Numerical Examples using Nonconvex Model

The nonconvex modification of the inpainting model discussed in Section 5.4.2 is tested on two example images, an image of grass and the brick wall image, both missing a large rectangular region in the center. In both examples, the weights end up being binary. Although the solutions are not a global minimizers of $F_{nc}(c)$, they look more natural than the global minimizers of the convex model.

For both examples, $\mu_{\Omega_o} = 1$, $\mu_{\Omega} = 1$, the correspondence term was multiplied by 1000, $s_{\Omega_o} = 100000$, $s_{\Omega} = 10000$ and $s_D = 100$.

The brick example in Figure 5.5 was computed in 400 iterations. For the first 200 iterations we set $\alpha = 1000$, $\delta = .001$ and $\gamma = \frac{.01}{2\alpha}$. For the last 200 iterations we set $\alpha = 100$, $\delta = .0001$ and $\gamma = \frac{.05}{2\alpha}$.

The grass example in Figure 5.6 was computed in 700 iterations. Similar to the brick example, for the first 500 iterations we set $\alpha = 1000$, $\delta = .001$ and $\gamma = \frac{.01}{2\alpha}$. For the last 200 iterations we set $\alpha = 100$, $\delta = .0001$ and $\gamma = \frac{.05}{2\alpha}$.

These parameters were not exhaustively optimized and better parameter selections may well lead to improved performance of the model.

Interestingly, the computed minimizers of F_{nc} are demonstrably not global minimizers. Even the global minimizers of F for the same examples have lower energy in terms of F_{nc}. The fact that the computed minimizers of F_{nc} lead to better solutions suggests that perhaps we shouldn't be looking for a global

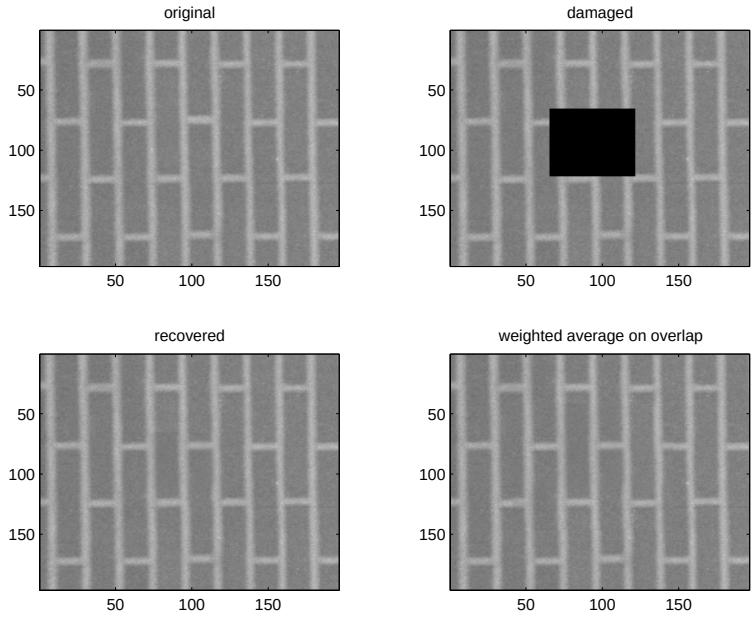


Figure 5.5: Inpainting brick wall using the nonconvex model with 45×45 patches

minimizer of that functional. Since the computed minimizers of F_{nc} are binary for the examples tested, this indicates that we may be more interested in computing minimizers of F subject to an additional constraint that restricts c to be binary. Our procedure for minimizing F_{nc} may be a practical approach for approximating solutions to that nonconvex problem.

5.5 Conclusions and Future Work

The proposed convex model for nonlocal inpainting can successfully be applied to simple images with repeating structure like the brick wall example. It's also a very good example of the efficiency of PDHGMp for large scale problems. Despite the high dimensionality of the model, the PDHGMp method is a practical means of solving it. However, in general the convex model tends to involve averaging

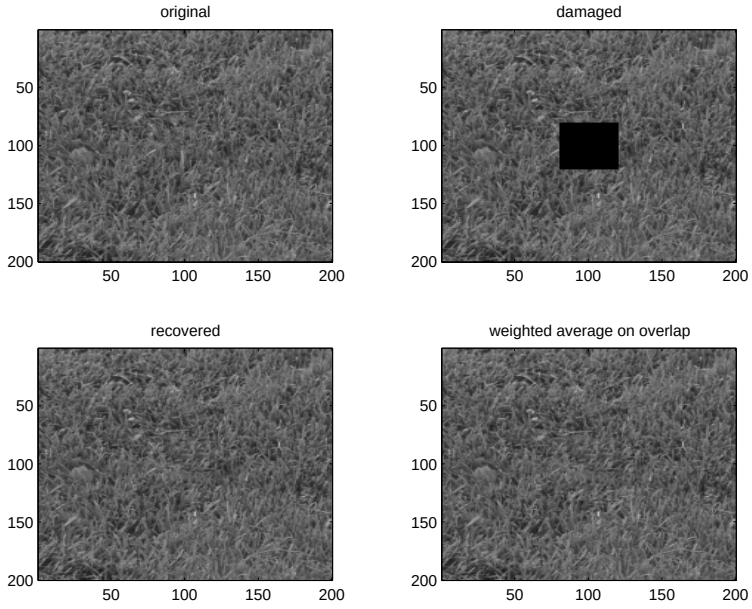


Figure 5.6: Inpainting grass using the nonconvex model with 15×15 patches

together too many known patches because this helps the unknown patches agree where they overlap. For images with repeating structure, the global minimizer prefers to at least take averages over similar patches to obtain better agreement at the boundary. But, when boundary agreement is difficult, as in more complicated images, the global minimizer can involve far too much averaging and yield visually poor solutions. Adding the nonconvex sparsifying term proved to be a fairly successful remedy to this problem, but the benefit of having a convex model was then lost.

Future work will involve investigating whether it's possible to achieve the sparser solutions while still staying in a convex framework. It may help to modify the strength of the data fidelity term depending on the distance to the boundary. This idea is used to improve inpainting results in [ACS09] via the introduction of a 'confidence mask.' It's also worth studying how to put the nonconvex model and

its numerical solution in a better theoretical framework. Of particular interest is the relationship between the binary computed minimizers of F_{nc} and minimizers of F with c constrained to be binary.

It may be possible to smooth the transition from the recovered solution on Ω to the known portion of the image on Ω^c by relaxing the $u \in S$ constraint. To accomplish this, Ω_u could be enlarged to cover the entire image, and $g_S(u)$ in (5.1) could be replaced with a quadratic penalty defined on Ω_o of the form $\frac{1}{2\eta}\|\mathcal{X}_{\Omega_o} \cdot (u - h_0)\|_F^2$. By weighting the term $\frac{1}{2\eta}\|\mathcal{X}_{\Omega_o} \cdot (u - h_0)\|_F^2$ more heavily away from Ω, it should still effectively act as the constraint $g_S(u)$ away from Ω and yet facilitate a smoother transition to the recovered image near Ω. It would also be interesting to combine this approach with a geometry inpainting model such as Euler elastica inpainting applied only near the boundary, similar to how texture synthesis and Euler elastica inpainting were combined in [Ni08].

Other modifications to consider are multiresolution approaches and expanding Ω_p to include patches that overlap the inpainting region. Multiresolution techniques, while doable, would be considerably more complicated to implement. Expanding Ω_p would make it possible to take weighted averages of patches which themselves depend on the unknown solution. This would allow the model to be more generally applicable, but it would also complicate the update for u and make the overall functional nonconvex.

References

[ACS09] P. Arias, V. Caselles, and G. Sapiro. "A variational framework for non-local image inpainting." In *Proc. of EMMCVPR*. Springer, 2009.

[ALM08] J. Aujol, S. Ladjal, and S. Masnou. "Exemplar-based inpainting from a variational point of view." Technical report, 2008. http://hal.archives-ouvertes.fr/docs/00/39/20/11/PDF/R09018.pdf.

[BBC] S. Becker, J. Bobin, and E. J. Candes. "NESTA: A Fast and Accurate First-Order Method for Sparse Recovery." http://arxiv.org/PS_cache/arxiv/pdf/0904/0904.3367v1.pdf.

[BBP04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. *Computer Vision - ECCV 2004*, volume 3024 of *LNCS*, chapter High Accuracy Optical Flow Estimation Based on a Theory for Warping, pp. 25–36. Springer, 2004.

[BCB09] E. Brown, T.F. Chan, and X. Bresson. "Convex Formulation and Exact Global Solutions for Multi-phase Piecewise Constant Mumford-Shah Image Segmentation." Technical report, 2009. UCLA CAM Report [09-66].

[Ber99] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

[Br73] H. Brézis. "Opérateurs Miximaux Monotones et Semi-Groupes de Contractions dans les Espaces de Hilbert." 1973.

[BT89] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation*. Prentice Hall, 1989.

[BT09] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems." *SIAM J. Imaging Sciences*, **2**:183–202, 2009.

[BYT09] E. Bae, J. Yuan, and X. Tai. "Global Minimization for Continuous Multiphase Partitioning Problems Using a Dual Approach." Technical report, 2009. UCLA CAM Report [09-75].

[CCN09] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, and T. Pock. "An introduction to Total Variation for Image Analysis." 2009. http://hal.archives-ouvertes.fr/docs/00/43/75/81/PDF/preprint.pdf.

[CCS08] J. F. Cai, R. H. Chan, and Z. Shen. "A Framelet-based Image Inpainting Algorithm." *Applied and Computational Harmonic Analysis*, **24**:131–149, 2008.

[CDS98] S. Chen, D. Donoho, and M. Saunders. "Atomic Decomposition by Basis Pursuit." *SIAM Journal on Scientific Computing*, **20**:33–61, 1998.

[CE04] T. F. Chan and S. Esedoglu. "Aspects of Total Variation Regularized L^1 function approximation." Technical report, 2004. UCLA CAM Report [04-07.

[CEN06] T. F. Chan, S. Esedoglu, and M. Nikolova. "Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models." *SIAM J. Appl. Math*, **66**(5):1632–1648, 2006.

[CGM99] T. F. Chan, G. H. Golub, and P. Mulet. "A nonlinear primal dual method for total variation based image restoration." *SIAM J. Sci. Comput.*, **20**, 1999.

[Cha04] A. Chambolle. "An Algorithm for Total Variation Minimization and Applications." *Journal of Mathematical Imaging and Vision*, pp. 89–97, 2004.

[CR05] E. Candes and J. Romberg. "Practical Signal Recovery from Random Projections." *IEEE Trans. Signal Processing*, 2005.

[CRT05] E. Candes, J. Romberg, and T. Tao. "Stable signal recovery from incomplete and inaccurate measurements." *Comm. Pure Appl. Math.*, **59**:1207–1223, 2005.

[CS05] T.F. Chan and J. Shen. "Variational Image Inpainting." *Comm. Pure Applied Math*, **58**:579–619, 2005.

[CSZ06] T. F. Chan, J. Shen, and H. Zhou. "Total Variation Wavelet Inpainting." *Journal of Mathematical Imaging and Vision*, 2006.

[CT94] G. Chen and M. Teboulle. "A Proximal-Based Decomposition Method for Convex Minimization Problems." *Mathematical Programming*, **64**:81–101, 1994.

[CV01] T.F. Chan and L.A. Vese. "Active Contours Without Edges." *IEEE Trans. Image Process*, **10**:266–277, 2001.

[CW06] P. Combettes and W. Wajs. "Signal Recovery by Proximal Forward-Backward Splitting." *Multiscale Modelling and Simulation*, 2006.

[DDM04] I. Daubechies, M. Defrise, and C. De Mol. "An Iterative Threshilding Algorithm for Linear Inverse Problems with a Sparsity Constraint." *Comm. Pure and Applied Math*, **57**, 2004.

[DR56] J. Douglas and H. H. Rachford. "On the Numerical Solution of Heat Conduction Problems in Two and Three Space Variables." *Transactions of the American mathematical Society*, **82**:421–439, 1956.

[DSC03] L. Demanet, B. Song, and T.F. Chan. "Image inpainting by correspondence maps: a deterministic approach." In *In Proc. VLSM, Nice,*, 2003.

[EB92] J. Eckstein and D. Bertsekas. "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators." *Mathematical Programming*, **55**, 1992.

[Eck89] J. Eckstein. *Splitting Methods for Monotone Operators with Applications to Parallel Optimization*. PhD thesis, Massachusetts Institute of Technology, Dept. of Civil Engineering, 1989.

[Eck93] J. Eckstein. "Nonlinear Proximal Point Algorithms Using Bregman Functions, with Applications to Convex Programming." *Mathematics of Operations Research*, **18**(1), 1993.

[EL99] A.A. Efros and T.K. Leung. "Texture Synthesis by Non-Parametric Sampling." *Computer Vision, IEEE International Conference on*, **2**:1022–1038, 1999.

[ELB08] A. Elmoataz, O. Lezoray, and S. Bougleux. "Nonlocal Discrete Regularization on Weighted Graphs: A framework for Image and Manifold Processing." *IEEE*, **17**(7), July 2008.

[Ess09] E. Esser. "Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman." Technical report, 2009. UCLA CAM Report [09-31].

[Ess10] E. Esser. "A Convex Model for Image Registration." Technical report, 2010. UCLA CAM Report [10-04].

[ET99] I. Ekeland and R. Temam. *Convex Analysis and Variational Problems*, volume 28 of *Classics in Applied Mathematics*. SIAM, 1999.

[EZC09] E. Esser, X. Zhang, and T. F. Chan. "A General Framework for a Class of First Order Primal-Dual Algorithms for TV Minimization." Technical report, 2009. UCLA CAM Report [09-67] (submitted to SIIMS).

[FG83] M. Fortin and R. Glowinski, editors. *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems.* Trans-Inter-Scientia, 1983.

[Gab79] D. Gabay. *Methodes numeriques pour l'optimisation non-lineaire.* PhD thesis, Universite Pierre et Marie Curie, 1979.

[Gab83] D. Gabay. *Augmented Lagrangian Methods: Application to the Solution of Boundary-Value Problems*, chapter Applications of the Method of Multipliers to Variational Inequalities. Trans-Inter-Scientia, 1983.

[GBO09] T. Goldstein, X. Bresson, and S. Osher. "Global Minimization of Markov Random Fields with Applications to Optical Flow." Technical report, 2009. UCLA CAM Report [09-77].

[GLN08] X. Guo, F. Li, and M. K. Ng. "A Fast l_1-TV Algorithm for Image Restoration." *SIAM J. Sci. Comput. (To Appear)*, 2008. http://www.math.hkbu.edu.hk/ICM/pdf/08-13.pdf.

[GM75] R. Glowinski and A. Marrocco. "Sur lapproximation par elements finis dordre un, et la resolution par penalisation-dualite dune classe de problemes de Dirichlet nonlineaires." **R-2**:41–76, 1975.

[GM76] D. Gabay and B. Mercier. "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations." *Comp. Math. Appl.*, **2**:17–40, 1976.

[GO09] T. Goldstein and S. Osher. "The Split Bregman Algorithm for L1 Regularized Problems." *SIAM Journal on Imaging Sciences*, **2**(2):323–343, 2009. UCLA CAM Report [08-29].

[GT89] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics.* SIAM, 1989.

[GY05] D. Goldfarb and W. Yin. "Second-order Cone Programming Methods for Total Variation-Based Image Restoration." *SIAM J. Sci. Comput.*, **27**(2):622–645, 2005.

[Hes69] R. M. Hestenes. "Multiplier and Gradient Methods." *Journal of Optimization Theory and Applications*, **4**:303–320, 1969.

[HJ85] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[HS81] B. Horn and B. Schunck. "Determining Optical Flow." *Artificial Intelligence*, **17**:185–203, 1981.

[HYZ07] E. Hale, W. Yin, and Y. Zhang. "A Fixed-Point Continuation Method for l_1-Regularized Minimization with Applications to Compressed Sensing." 2007. CAAM Technical Report TR07-07.

[Ish03] H. Ishikawa. "Exact Optimization for Markov Random Fields with Convex Priors." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**:1333–1336, 2003.

[LC01] M. Lefébure and L. Cohen. "Image Registration, Optical Flow and Local Rigidity." *Journal of Mathematical Imaging and Vision*, **14**:131–147, 2001.

[LGD09] T. Lin, C. Le Guyader, I. Dinov, P. Thompson, A. Toga, and L. Vese. "A Landmark-Based Image Registration Model using a Nonlinear Elasticity Smoother for Mapping Mouse Atlas to Gene Expression Data." Technical report, 2009. UCLA CAM Report [09-51].

[LM79] L. P. Lions and B. Mercier. "Algorithms for the Sum of Two Nonlinear Operators." *SIAM Journal on Numerical Analysis*, **16**(6):964–979, 1979.

[LPS03] F. Larsson, M. Patriksson, and A.-B. Stromberg. "On the convergence of conditional ϵ-subgradient methods for convex programs and convex-concave saddle-point problems." *European J. Oper. Res.*, **151**(3):461–473, 2003.

[Min62] G. J. Minty. "Monotone (Nonlinear) Operators in Hilbert Space." *Duke Mathematics Journal*, **29**:341–346, 1962.

[Mor65] J. J. Moreau. "Proximité et dualité dans un espace hilbertien." *Bull. Soc. Math. France*, **93**:273–299, 1965.

[Nes05] Y. Nesterov. "Smooth Minimization of Non-Smooth Functions." *Math. Program.*, (103):127–152, 2005.

[Nes07] Y. Nesterov. "Dual extrapolation and its applications to solving variational inequalities and related problems." *Math. Program.*, **119**:319–344, 2007.

[Ni08] K. Ni. "Variational PDE-Based Image Segmentation and Inpainting with Applications in Computer Graphics.", 2008. UCLA CAM Report [08-39].

[Pas79] G. B. Passty. "Ergodic Convergence to a Zero of the Sum of Monotone Operators in Hilbert Space." *Journal of Mathematical Analysis and Applications*, **72**:383–390, 1979.

[PCB09] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. "An Algorithm for Minimizing the Mumford-Shah Functional." In *In ICCV Proceedings*. Springer, 2009.

[Pop80] L. Popov. "A Modification of the Arrow-Hurwicz Method for Search of Saddle Points." *Math. Notes*, **28(5)**:845–848, 1980.

[PR55] D. H. Peaceman and H. H. Rachford. "The Numerical Solution of Parabolic Elliptic Differential Equations." *SIAM Journal on Applied Mathematics*, **3**:28–41, 1955.

[PSG08] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. "A Convex Formulation of Continuous Multi-Label Problems." In *ECCV, Proceedings of the 10th European Conference on Computer Vision*, pp. 792–805. Springer-Verlag, 2008.

[Roc70] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

[Roc76] R. T. Rockafellar. "Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming." *Mathematics of Operations Research*, **1**(2):97–116, 1976.

[ROF92] L. Rudin, S. Osher, and E. Fatemi. "Nonlinear Total Variation Based Noise Removal Algorithms." *Physica D*, **60**:259–268, 1992.

[Set09] S. Setzer. "Split Bregman Algorithm, Douglas-Rachford Splitting and Frame Shrinkage." **5567**:464–476, 2009.

[SKR85] N. Z. Shor, K. C. Kiwiel, and A. Ruszcyǹski. *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag New York, Inc., 1985.

[SW75] G. Stephanopoulos and A. W. Westerberg. "The Use of Hestenes' Method of Multipliers to Resolve Dual Gaps in Engineering System Optimization." *Journal of Optimization Theory and Applications*, **15**(3), 1975.

[TGS06] H. Tagare, D. Groisser, and O. Skrinjar. "A geometric theory of symmetric registration." In *In Proceedings of CVPR*. IEEE, 2006.

[Tse91] P. Tseng. "Applications of a Splitting Algorithm to Decomposition in Convex Programming and Variational Inequalities." *SIAM J. Control and Optimization*, **29**(1):119–138, 1991.

[Tse08] P. Tseng. "On Accelerated Proximal Gradient Methods for Convex-Concave Optimization." *SIAM Journal on Optimization*, 2008. Submitted.

[Tse09] P. Tseng, 2009. Private communication.

[TW09] X. Tai and C. Wu. "Augmented Lagrangian Method, Dual Methods and Split Bregman Iteration for ROF Model." Technical report, 2009. UCLA CAM Report [09-05].

[WAB07] P. Weiss, G. Aubert, and L. Blanc-Féraud. "Efficient Schemes for Total Variation Minimization Under Constraints in Image Processing." 2007. INRIA, No. 6260.

[WT09] C. Wu and X. Tai. "Augmented Lagrangian Method, Dual Methods, and Split Bregman Iteration for ROF, Vectorial TV, and High Order Models." Technical report, 2009. UCLA CAM Report [09-76].

[WYZ07] Y. Wang, W. Yin, and Y. Zhang. "A Fast Algorithm for Image Deblurring with Total Variation Regularization." Technical report, 2007. UCLA CAM Report [07-22].

[YOG08] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. "Bregman Iterative Algorithms for l_1-Minimization with Applications to Compressed Sensing." *SIAM J. Imaging Science*, **1**:142–168, 2008. UCLA CAM Report [07-37].

[YZY09] J. Yang, Y. Zhang, and W. Yin. "An Efficient TVL1 Algorithm for Deblurring Multichannel Images Corrupted by Impulsive Noise." *SIAM J. Sci. Comput.*, **31**:2842–2865, 2009.

[ZBB09] X. Zhang, M. Burger, X. Bresson, and S. Osher. "Bregmanized Nonlocal Regularization for Deconvolution and Sparse Reconstruction." Technical report, 2009. UCLA CAM Report [09-03].

[ZBO09] X. Zhang, M. Burger, and S. Osher. "A Unified Primal-Dual Algorithm Framework Based on Bregman Iteration." Technical report, 2009. UCLA CAM Report [09-99].

[ZC08] M. Zhu and T. F. Chan. "An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration." Technical report, 2008. UCLA CAM Report [08-34].

[ZC09] X. Zhang and T.F. Chan. "Wavelet Inpainting by Nonlocal Total Variation." Technical report, 2009. UCLA CAM Report [09-64].

[ZGF08] C. Zach, D. Gallup, J.-M. Frahm, , and M. Niethammer. "Fast global labeling for real-time stereo using multiple plane sweeps." In *Proceedings of Vision, Modeling and Visualization Workshop (VMV)*, 2008.

[ZPB07] C. Zach, T. Pock, and H. Bischof. "A Duality Based Approach for Realtime TV-L^1 Optical Flow." In *Proceedings of the 29th DAGM Symposium on Pattern Recognition*. Springer LNCS, 2007.

[ZWC08] M. Zhu, S. J. Wright, and T. F. Chan. "Duality-Based Algorithms for Total-Variation-Regularized Image Restoration." *Comput. Optim. Appl.*, 2008.