UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO
-
UNIVERSITÉ GRENOBLE ALPES

On some aspects of inverse problems in image processing

Miguel Antunes Dias Alfaiate Simões

Supervisor:	Doctor Luís Henrique Martins Borges de Almeida
Co-Supervisor:	Doctor José Manuel Bioucas Dias
Co-Supervisor:	Doctor Jocelyn Chanussot

Thesis approved in public session to obtain the PhD Degree in
Electrical and Computer Engineering

Jury final classification: Pass with Distinction and Honour

2017

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO
-
UNIVERSITÉ GRENOBLE ALPES

On some aspects of inverse problems in image processing

Miguel Antunes Dias Alfaiate Simões

Supervisor: Doctor Luís Henrique Martins Borges de Almeida
Co-Supervisor: Doctor José Manuel Bioucas Dias
Co-Supervisor: Doctor Jocelyn Chanussot

Thesis approved in public session to obtain the PhD Degree in
Electrical and Computer Engineering

Jury final classification: Pass with Distinction and Honour

Jury

Chairperson: Doctor Mário Alexandre Teles de Figueiredo, Instituto Superior Técnico, Universidade de Lisboa

Members of the Committee:

Doctor Nicolas Dobigeon, Université Fédérale de Toulouse Midi-Pyrénées, France

Doctor Luís Henrique Martins Borges de Almeida, Instituto Superior Técnico, Universidade de Lisboa

Doctor Paulo Jorge dos Santos Gonçalves Ferreira, Universidade de Aveiro

Doctor Mário Alexandre Teles de Figueiredo, Instituto Superior Técnico, Universidade de Lisboa

Doctor Pedro Manuel Quintas Aguiar, Instituto Superior Técnico, Universidade de Lisboa

Doctor Laurent Condat, CNRS, GIPSA-lab, Université Grenoble Alpes, France

Funding Institutions: Instituto de Telecomunicações (IT), GIPSA-lab, Fundação para a Ciência e a Tecnologia (FCT)

2017

On some aspects of inverse problems in image processing

Miguel Antunes Dias Alfaiate Simões

2017

Acknowledgements

This section has been omitted in the present version.

Abstract

This work is concerned with two image-processing problems, image deconvolution with incomplete observations and data fusion of spectral images, and with some of the algorithms that are used to solve these and related problems.

In image-deconvolution problems, the diagonalization of the blurring operator by means of the discrete Fourier transform usually yields very large speedups. When there are incomplete observations (e.g., in the case of unknown boundaries), standard deconvolution techniques normally involve non-diagonalizable operators, resulting in rather slow methods, or, otherwise, use inexact convolution models, resulting in the occurrence of artifacts in the enhanced images. We propose a new deconvolution framework for images with incomplete observations that allows one to work with diagonalizable convolution operators, and therefore is very fast. The framework is also an efficient, high-quality alternative to existing methods of dealing with the image boundaries, such as edge tapering.

The data-fusion problem of inferring a hyperspectral image with high spectral and spatial resolutions from a spatially-degraded hyperspectral image and a multispectral image retrieved from the same geographical area has been a subject of recent research. We formulate this problem as the minimization of a convex function containing two quadratic data-fitting terms and an edge-preserving regularizer. The regularizer, a form of vector total variation, promotes piecewise-smooth solutions with discontinuities aligned across the hyperspectral bands. We obtain an algorithm that outperforms the state of the art, as illustrated in a series of experiments.

The algorithms that are used to solve problems with sparsity-inducing regularizers are usually generic, in the sense that they do not take into account the sparsity of the solution in any particular way. However, methods such as the semismooth Newton and the active-set ones are able to take advantage of this sparsity to accelerate their convergence. We show how to extend these algorithms in different directions, and study their convergence in (possibly infinite-dimensional) real Hilbert spaces. Additionally, we discuss the use of second-order information in the alternating-direction method of multipliers when solving ℓ^2+regularizer minimization problems.

Kewwords: Deconvolution, incomplete observations, hyperspectral imaging, superresolution, data fusion, pansharpening, convex nonsmooth optimization, primal-dual optimization, alternating-direction method of multipliers (ADMM), semismooth Newton method, forward-backward method, monotone inclusion, averaged operator, variable metric.

Resumo

Esta tese aborda dois problemas de processamento de imagem, a desconvolução de imagem com observações incompletas e a fusão de dados de imagens espectrais. Aborda igualmente alguns dos algoritmos que são usados para resolver estes e outros problemas.

Em problemas de desconvolução de imagem, a diagonalização do operador de desfocagem através da transformada discreta de Fourier permite geralmente obter ganhos significativos de velocidade. Quando algumas das observações estão incompletas (por exemplo, no caso de as fronteiras serem desconhecidas), as técnicas de desconvolução clássicas envolvem normalmente operadores não diagonalizáveis, o que resulta em métodos bastante lentos, ou, então, usam modelos de convolução inexatos, o que resulta no aparecimento de efeitos espúrios nas imagens restauradas. Neste contexto, propõe-se na presente tese uma nova abordagem à desconvolução de imagens com observações incompletas. Esta abordagem permite trabalhar com operadores de convolução diagonalizáveis sendo, portanto, muito rápida. É igualmente uma alternativa eficiente e de alta qualidade a métodos existentes para lidar com as fronteiras de imagens como *edge tapering*.

O problema de fusão de dados, que diz respeito à produção de uma imagem hiperespectral de altas resoluções espacial e espectral a partir de uma imagem hiperespectral degradada no espaço e de uma imagem multiespectral, ambas adquiridas a partir da mesma área geográfica, tem sido alvo de atenção recente. Nesta tese, este problema foi formulado como a minimização de uma função convexa contendo dois termos quadráticos de ligação aos dados e um termo de regularização que preserva os contornos da imagem. O termo de regularização, que é uma forma vectorial do operador de variação total, promove soluções suaves por troços com as descontinuidades alinhadas ao longo das bandas hiperespectrais. É obtido um algoritmo cujo desempenho supera o de outros métodos, conforme ilustrado numa série de experiências.

Os algoritmos que são usados para resolver problemas com termos de regularização que impõem esparsidade são habitualmente genéricos, no sentido em que consideram a esparsidade da solução de nenhuma forma em particular. No entanto, métodos como os de Newton semi-suave e de conjunto activo são capazes de tirar partido dessa esparsidade para acelerar a sua convergência. É apresentada nesta tese uma forma de estender estes algoritmos em diferentes direções e de estudar a sua convergência em espaços de Hilbert reais (possivelmente infinitos). Para além disto, é discutido o uso de informação de segunda ordem no método dos multiplicadores de direção alternada (ADMM) quando este é usado para resolver problemas de minimização do tipo ℓ^2+regularizador.

Palavras-chave: Desconvolução, observações incompletas, imagem hiperespectral, super-resolução, fusão de dados, *pansharpening*, otimização convexa não-suave, otimização primal-dual, método dos multiplicadores de direção alternada (ADMM), método de Newton semi-suave, método explícito-implícito, inclusão monotónica, operador médio, métrica variável.

Résumé

Cette thèse s'intéresse à deux problèmes de traitement d'image, la déconvolution d'image avec des observations incomplètes et la fusion d'images spectrales. Elle s'intéresse aussi à certains aspects algorithmiques nécessaires pour résoudre ces problèmes, et d'autres.

Dans les problèmes de déconvolution d'image, la diagonalisation de l'opérateur de flou par la transformation de Fourier discrète permet généralement des gains de vitesse considérables. Lorsqu'il y a des observations incomplètes (par exemple, dans le cas de frontières inconnues), les techniques de déconvolution classiques impliquent généralement des opérateurs non diagonalisables, entraînant des méthodes plutôt lentes ou, sinon, utilisent des modèles de convolution inexacts, ce qui entraîne l'apparition de défauts dans les images nettes. Dans ce contexte, on propose un nouveau cadre de déconvolution pour les images avec des observations incomplètes. Ce cadre permet de travailler avec des opérateurs de convolution diagonalisables et est donc très rapide. Il est aussi une alternative efficace et de haute qualité aux méthodes existantes de traitement des frontières d'une image.

Le problème de la fusion de données qui concerne la production d'une image hyperspectrale de hautes résolutions spatiale et spectrale à partir d'une image hyperspectrale dégradée dans l'espace et d'une image multispectrale extraites de la même zone géographique, a fait l'objet d'une attention récente. Dans cette thèse, ce problème est formulé comme la minimisation d'une fonction convexe contenant deux termes quadratiques d'attache aux données et un terme de régularisation qui préserve les contours de l'image. Le terme de régularisation, qui est une forme vectorielle de l'opérateur de variation totale, favorise des solutions lisses par morceaux avec les discontinuités alignées au travers des bandes hyperspectrales. La méthode proposée est validée sur différents jeux de données et comparée à d'autres méthodes de l'état de l'art, par rapport auxquelles elle produit des performances supérieures.

Les algorithmes qui sont utilisés pour résoudre des problèmes avec des termes de régularisation imposant de la parcimonie sont habituellement génériques, au sens qu'ils ne tiennent pas compte de la parcimonie de la solution d'une manière particulière. Cependant, des méthodes telles que l'algorithme de Newton semi-lisse et par ensemble actif peuvent profiter de cette parcimonie pour accélérer leur convergence. Une façon d'étendre ces algorithmes dans des directions différentes et d'étudier leur convergence dans des espaces hilbertiens réels (éventuellement de dimension infinie) est présentée dans cette thèse. Par ailleurs, nous discutons de l'utilisation d'information de second ordre dans la méthode des multiplicateurs de direction alternée (ADMM) lors de la résolution de problèmes de minimisation du type ℓ^2+regularizateur.

Mots-clés : Déconvolution, observations incomplètes, imagerie hyperspectrale, super-résolution, fusion de données, *pansharpening*, optimisation convexe nonlisse, optimisation primale-duale, méthode des multiplicateurs de direction alternée (ADMM), méthode de Newton semi-lisse, méthode explicite-implicite, inclusions monotones, opérateur moyenné, métrique variable.

Preface

This dissertation addresses three topics that are related to inverse problems in image processing: (1) image deconvolution with incomplete observations, (2) data fusion of spectral images, and (3) algorithms that are used to solve (among others) inverse problems in image processing. After providing some background, we start by considering the problem of image deconvolution with incomplete observations. We propose a new framework that can be used to deconvolve images with pixels that are not observed. The algorithms obtained from this framework are very fast, namely because we are allowed to work with diagonalizable convolution operators. We alternate the estimation between the non-observed pixels and the sharp image. The framework can also be used to obtain an efficient, high-quality alternative to existing methods of dealing with the image boundaries, such as edge tapering. It can be used with any fast deconvolution method, e.g., a fast Fourier Transform (FFT)-based one. We give an example in which a state-of-the-art method that assumes periodic boundary conditions is extended, through the use of this framework, to unknown boundary conditions. Furthermore, we propose a specific implementation of this framework that is based on the alternating-direction method of multipliers (ADMM), and provide a proof of convergence for the resulting algorithm. This implementation can be seen as a “partial” ADMM, since not all variables are dualized. We report experimental comparisons with other primal–dual methods, where the proposed one performed at the level of the state of the art. Four different kinds of applications were tested: deconvolution, deconvolution with inpainting, superresolution, and demosaicing, all with unknown boundary conditions.

We then consider the problem of fusing hyperspectral images (HSIs) with multispectral images (MSIs). In general, HSIs have high spectral resolution and low spatial resolution, whereas MSIs have low spectral and high spatial resolutions. Due to the increasing availability of HSIs and MSIs retrieved from the same geographical area, the problem of inferring images that combine the high spectral and high spatial resolutions of HSIs and MSIs, respectively, has been the focus of recent research. We formulate this problem as the minimization of a convex function containing two quadratic data-fitting terms and an edge-preserving regularizer. The data-fitting terms account for blur, different resolutions, and additive noise. The regularizer, a form of vector total variation, promotes piecewise-smooth solutions with discontinuities aligned across the hyperspectral bands. The resulting optimization problem is hard to solve due to the presence of a downsampling operator accounting for the different spatial resolutions, to the non-quadratic and non-smooth nature of the regularizer, and to the very large size of the HSI to be estimated. We deal with these difficulties by exploiting the fact that HSIs typically lie in low-dimensional subspaces and by tailoring an instance of ADMM to this optimization problem. The spatial and the spectral blur operators linked, respectively, with the HSI and MSI acquisition processes are also estimated. We obtain an effective algorithm that outperforms the state of the art, as illustrated in a series of experiments. In addition, we also discuss the contributions made by the writer in two different, though related, works. The first contribution is concerned with an issue present in the pansharpening algorithms based on the detail-injection model. Such algorithms are usually comprised of two steps: extraction of the spatial details of the panchromatic image (PAN), and their subsequent injection into the MSI. In order to obtain these spatial details, one may use a procedure that requires the application of a particular low-pass filter to the PAN. We propose a new algorithm for estimating this low-pass filter that only makes use of the observed MSI and PAN. The second contribution is concerned with the algorithms used to fuse hyperspectral and multispectral images that take advantage of the fact that HSIs usually lie in a low-dimensional space. These methods perform at the level of the state of the art if the HSIs actually lie in such a space. However, if the dimensionality of this space is not low, in the sense that it is larger than the number of multispectral bands, the performance of these methods is not satisfactory. We propose a local approach to cope with this difficulty by exploiting the fact that real-world HSIs are locally low rank, which is a consequence of the fact that, in a small spatial neighborhood, the number of different materials is typically small.

Finally, we consider algorithms used to solve problems involving sparsity-inducing regularizers. Typically, these are generic, in the sense that they do not take into account the sparsity of the solution in any particular way. However, some of them, such as semismooth Newton methods, are able to take advantage of this sparsity to accelerate their convergence. We show how to extend these algorithms in different directions, and study the convergence of the resulting algorithms in real Hilbert spaces. We base our analysis on a variation on the well-known Krasnosel'skiĭ–Mann scheme, and show that these methods are a particular case of this variation. Additionally, we discuss the use of second-order information in ADMM when used to solve ℓ^2-regularized minimization problems. In particular, we compare and contrast ADMM with a particular instance of a variable–metric primal–dual method.

Contributions

The original contributions discussed in this work can be listed as follows:

1. a new deconvolution framework for images with incomplete observations;
2. a specific implementation of this framework based on ADMM and a proof of the convergence of this implementation;
3. a data-fusion algorithm that combines the high spectral and high spatial resolutions of hyperspectral images and multispectral images, respectively;
4. a method that estimates the spatial and the spectral blur operators linked with the acquisition processes of the hyperspectral images and multispectral sensors, respectively;

5. the study of an interpretation of semismooth Newton methods as a variation on the well-known Krasnosel'skiĭ–Mann scheme and subsequent analysis of extensions of these methods; and

6. a discussion showing that ADMM shares some similarities with a second-order primal–dual method when used to solve ℓ^2-regularized minimization problems.

Additionally, we also briefly describe the original contributions of the writer in the context of work done in collaboration with other authors. These contributions are

7. a method that estimates the low-pass filter to be used in pansharpening algorithms based on the detail-injection model and that makes use of only the observed MSI and PAN, and

8. a local approach to the algorithms used to fuse HSIs and MSIs, which takes advantage of the fact that HSIs usually lie in a low-dimensional space.

The contributions indicated by points 1-4, 7, and 8 have been the subject of publication, whereas the ones indicated by points 5 and 6 still have not. The publications, in chronological order, are the following:

[1] G. A. Licciardi, M. A. Veganzones, M. Simões, J. Bioucas-Dias, and J. Chanussot, "Super-resolution of hyperspectral images using local spectral unmixing," in *IEEE Workshop Hyperspectral Image Signal Proces.: Evolution Remote Sens.*, Lausanne, Switzerland, June 2014;

[2] M. A. Veganzones, M. Simões, G. Licciardi, J. Bioucas-Dias, and J. Chanussot, "Hyperspectral super-resolution of locally low rank images from complementary multisource data," in *IEEE Int. Conf. Image Processing*, Paris, France, Oct. 2014, pp. 703–707;

[3] M. Simões, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "Hyperspectral image superresolution: An edge-preserving convex formulation," in *IEEE Int. Conf. Image Processing*, Paris, France, Oct. 2014, pp. 4166–4170;

[4] G. Vivone, M. Simões, M. Dalla Mura, R. Restaino, J. Bioucas-Dias, G. Licciardi, and J. Chanussot, "Pansharpening based on semiblind deconvolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 1997–2010, Apr. 2015;

[5] M. Simões, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "A convex formulation for hyperspectral image superresolution via subspace-based regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3373–3388, June 2015;

[6] L. Loncan, L. B. Almeida, J. Bioucas-Dias, X. Briottet, J. Chanussot, N. Dobigeon, S. Fabre, W. Liao, G. Licciardi, M. Simões, J.-Y. Tourneret, M. A. Veganzones, G. Vivone, Q. Wei, and N. Yokoya, "Comparison of nine hyperspectral pansharpening methods," in *IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, July 2015;

[7] ——, “Hyperspectral pansharpening: A review,” *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 27–46, Sept. 2015;

[8] M. A. Veganzones, M. Simões, G. Licciardi, N. Yokoya, J. Bioucas-Dias, and J. Chanussot, “Hyperspectral super-resolution of locally low rank images from complementary multisource data,” *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 274–288, Jan. 2016;

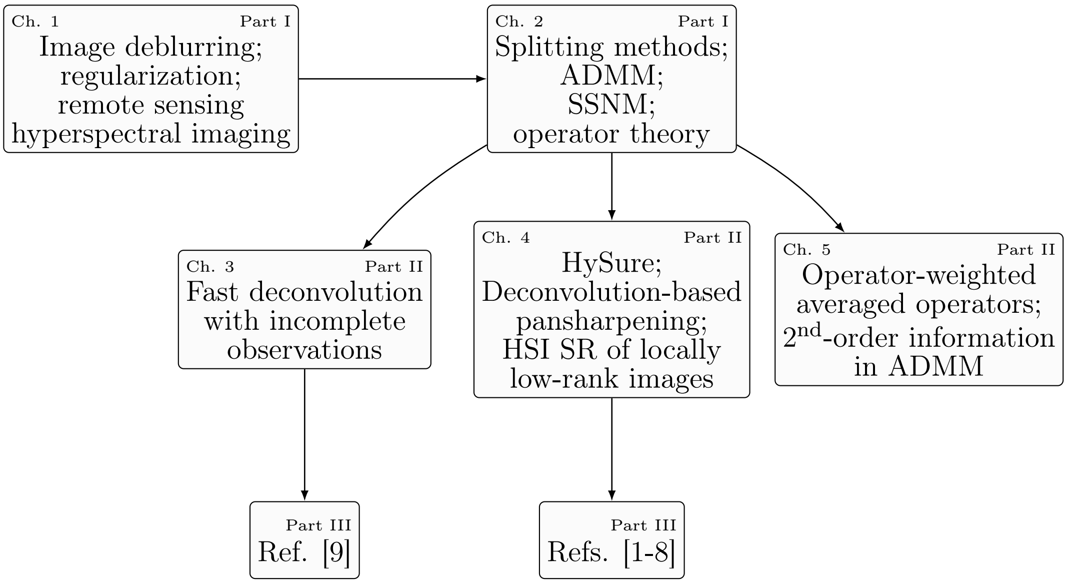
[9] M. Simões, L. B. Almeida, J. Bioucas-Dias, and J. Chanussot, “A framework for fast image deconvolution with incomplete observations,” *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5266–5280, Nov. 2016.

Copies of these publications are included at the end of this work.

Outline

The structure of this work is as follows. In Part I, we provide some background. In particular, in Chapter 1, we discuss some characteristics of inverse problems by focusing on the problem of image deblurring. We also discuss regularization problems, and give a brief overview of remote-sensing hyperspectral imaging. In Chapter 2, we list some of the algorithms that are used to solve imaging and other large-scale problems. We discuss smooth and nonsmooth minimization, as well as splitting methods and semismooth Newton ones. We then introduce some notions of operator theory. In Part II, we describe, in some detail, the original contributions of this work. More specifically, in Chapter 3, we discuss the contributions numbered 1 and 2, above. Then, in Chapter 4, we discuss the contributions numbered 3, 4, 7, and 8. Finally, in Chapter 5, we discuss the contributions numbered 5 and 6. Part III contains the publications listed in the previous section.

The figure below schematizes the structure of this work.


Ch. 1
Part I
Image deblurring; regularization; remote sensing hyperspectral imaging
Ch. 2
Part I
Splitting methods; ADMM; SSNM; operator theory
Ch. 3
Part II
Fast deconvolution with incomplete observations
Ch. 4
Part II
HySure; Deconvolution-based pansharpening; HSI SR of locally low-rank images
Ch. 5
Part II
Operator-weighted averaged operators; 2nd-order information in ADMM
Part III
Ref. [9]
Part III
Refs. [1-8]

where we have used the following acronyms:
ADMM: alternating-direction method of multipliers,
SSNM: semismooth Newton method,
HySure: hyperspectral superresolution method, and
HSI SR: hyperspectral image superresolution.

Contents

List of Figures

List of Algorithms

Part I.

Background

1. An approach to digital imaging inverse problems

This chapter describes, in broad strokes, an approach that is widely used to solve some problems in imaging, namely the ones that assume the data to be generated by a given stochastic model.[1] This approach consists in using the framework of inverse problems. In what follows, we briefly discuss the conceptual advantages of this approach by using image deconvolution as an example. We also give a brief overview of remote-sensing hyperspectral imaging.

Contents

1.1. Introduction

Digital images have become ubiquitous in our modern society. They are not only used to capture our life's most memorable moments, but find numerous applications in astronomy, medicine, geoscience, engineering, and other fields. Since the advent of digital imaging in the 60s, it has become progressively cheaper to capture and store a large number of images. These images can be processed to enhance some of their characteristics—such as their resolution or sharpness—or to facilitate their storage and transmission. In addition, digital images can be generated artificially, both for human and for machine perception. The use of processing methods involving these images is referred to as *digital image processing*. Many of these methods can be framed as approaches to tackle *inverse problems*, which we define next.

Consider a physical system that is observed by some sensors producing a set of *observations* (or measurements). Assume that we are able to mathematically model this

[1] This is in contrast to the approach that treats the data-generating mechanism as a "black box". For a discussion on the merits of these two approaches, we refer the reader to Breiman [10].

system, and that the resulting model has a set of unknowns—usually termed *original* (or *target*) data—reflecting the underlying physical reality. In science and engineering, it is rare to have knowledge of both the original data and the observations. Often, one tries to find estimates of one from the other. The problem of finding the observations from the original data is called the *direct problem* (or forward problem), whereas finding the original data from the observations is called the *inverse problem.* As an example of a physical system, consider a digital camera acquiring visual information of its surroundings in the form of images (these are the aforementioned observations). Frequently, the images appear to be *noisy*, suggesting that they were degraded by some mechanism. The imaging system can be modeled, under certain assumptions, as follows: the noisy image is assumed to be given by the sum of two terms, where one is the noise itself, which depends on the camera, and the other is a non-noisy version of the observed image, which corresponds to the original image. Obtaining an estimate of this image is, then, the inverse problem, which, in this context, is often called *image denoising.* In contrast, the noisy image is the solution to the direct problem, given knowledge of the camera and the original image.

Another example of an inverse problem is *image deblurring*, i.e., the recovery of sharp images from blurred ones. Consider again the scenario discussed in the previous paragraph. An additional image degradation can be due to depth-of-field effects and defocusing. These degradation mechanisms produce images that are not only noisy but also blurred. Other phenomena can occur that also result in a decrease in image sharpness, e.g., camera motion or the propagation of light through the atmosphere. A model for a camera that produces blurred images is discussed in Section 1.2. In these problems, the model depends on a number of parameters that characterize the degradation mechanisms. For example, if we assume the noise to follow a certain distribution, examples of those parameters are the parameters that characterize this distribution.

Image processing techniques can be used to solve specific problems in all the fields that make use of digital images. One of these fields is remote sensing, the science of obtaining information about an object *remotely*, i.e., from a distance. In the case of earth remote sensing, this is done typically by using sensors on-board satellites or airplanes. Some of these sensors capture optical spectral images, which are sets of 2-D images representing the reflectance or radiance of an object across different bands of the electromagnetic (EM) spectrum. Each individual 2-D image corresponds to a spectral band. The number of 2-D images is related with the spectral resolution of the spectral image. Enhancing the resolution of these images, both in the spatial and spectral domain, is another example of an inverse problem. In this case, the system is usually modeled as a degradation process acting on a sharp spectral image. This degradation, which occurs both in the spectral and spatial domains, is a consequence of the characteristics of the sensor that was used to capture the image.

Optical spectral images are usually classified into two categories: multi- and hyperspectral images. The distinction between the two is not clear-cut, but multispectral images typically have around ten channels, whereas hyperspectral images have more than one hundred. The sensors used to capture each category of images have different characteristics, and there usually is a trade-off between their spectral and spatial resolutions. Typically, a multispectral sensor produces images with relatively high spatial and low spectral resolutions, whereas a hyperspectral sensor produces images with relatively high spectral and low spatial resolutions. When images of both categories are captured from the same scene, it is often of interest to combine them in order to obtain an image with both high spectral and high spatial resolutions. This combination, again, can be addressed within the framework of inverse problems. The system is usually modeled as two different degradations processes acting on a sharp spectral image: one corresponding to the hyperspectral sensor and the other to the multispectral one (this model is discussed in Section 4.1).

The main advantage of analyzing all the different problems described in the previous paragraphs as inverse problems lies in the fact that this can be done in a systematic way. Recall that the end goal of solving an inverse problem is to recover the original data from the observations. This is typically harder to do than the reverse. As an example, consider the inverse problem of image deblurring: it requires the estimation of not only a deblurred image but also of the parameters associated with the camera. It can then be helpful to devise ways of using additional information about the system under study. This can be done within the framework under discussion, as we show next.

When formulating an inverse problem, we usually need to address four different points. These are: (a) the selection of an observation model describing the underlying physical reality, (b) the establishment of a criterion quantifying how well the observations are described by the model, (c) the design of a way to incorporate any additional information about the parameters, if available, and (d) the selection of a computational approach to tackle the inverse problem. Point (a) is, in and of itself, a separate problem known as a model identification. Regarding points (b) and (c), the approach followed in this work considers the solution of an inverse problem to be the solution of an optimization problem, where some of the terms of its cost function are related to point (b), and others are related to point (c). Point (d), again in the case of considering the inverse problem as an optimization one, corresponds to the selection of the algorithm used to solve it.

In short, in order to solve inverse problems, and assuming that they are formulated as optimization problems, we need to at least consider the following points:

1. the observation *model* used to describe the physical system,

2. the *criterion* used to evaluate how well the model fits the available observations,

3. *additional information* about the desired solution, and

4. the *algorithm* used to solve the optimization problem.

Each of these points is currently a subject of intense research, and the methods that are proposed to tackle imaging problems need to always address them. In fact, differences between competing methods can be traced to differences in approach to these points. On the other hand, problems in very different domains may share the same approach to one (or more) of these points.

1.1.1. Notation

Image processing is at a crossroads of different fields, namely computer science, electrical engineering, and applied mathematics. The communities working on these fields tend to use different notations for the same objects. Since this work has contributions in the field of signal processing and in the field of convex analysis and optimization, our notation tries to maintain consistency with both fields. We follow the notation used by Bauschke and Combettes [11] when considering optimization algorithms in the context of operator theory, and we follow the so-called vector notation used in the image-processing community when dealing with digital images or other finite-dimensional objects. A list of symbols and notation is given on page xix.

In this work, calligraphic uppercase letters denote *Hilbert spaces*, as in $\mathcal{X}$, $\mathcal{V}$. All the spaces under consideration are real and might be infinite.[2] We denote the *scalar product* of a Hilbert space by $\langle \cdot, \cdot \rangle$ and the associated *norm* by $\| \cdot \|$. $2^{\mathcal{V}}$ denotes the *power* set of $\mathcal{V}$, i.e., the set of all subsets of $\mathcal{V}$. An *operator* (or mapping) $A : \mathcal{X} \to \mathcal{V}$ maps each point in $\mathcal{X}$ to a point in $\mathcal{V}$. A *set-valued* operator $A : \mathcal{X} \to 2^{\mathcal{V}}$ maps each element in $\mathcal{X}$ to a set in $\mathcal{V}$. Id denotes the *identity* operator. $\mathcal{L}(\mathcal{X}, \mathcal{V})$ denotes the space of *linear* operators from $\mathcal{X}$ to $\mathcal{V}$ and $\mathcal{B}(\mathcal{X}, \mathcal{V})$ denotes the space of *bounded linear* operators from $\mathcal{X}$ to $\mathcal{V}$. We set $\mathcal{L}(\mathcal{X}) \triangleq \mathcal{L}(\mathcal{X}, \mathcal{X})$ and $\mathcal{B}(\mathcal{X}) \triangleq \mathcal{B}(\mathcal{X}, \mathcal{X})$. Given an operator $A \in \mathcal{B}(\mathcal{X}, \mathcal{V})$, its *adjoint* A^* is the operator $A^* : \mathcal{V} \to \mathcal{X}$ such that for all $x \in \mathcal{X}$ and $u \in \mathcal{V}$, $\langle Ax, u \rangle = \langle x, A^* u \rangle$. $\mathcal{S}(\mathcal{X})$ denotes the space of self-adjoint bounded linear operators from $\mathcal{X}$ to $\mathcal{X}$, i.e., $\mathcal{S}(\mathcal{X}) \triangleq \{A \in \mathcal{B}(\mathcal{X}) \,|\, A = A^*\}$. Given two operators A, $B \in \mathcal{S}(\mathcal{X})$, the Loewner partial ordering on $\mathcal{S}(\mathcal{X})$ is defined by $A \succeq B \Leftrightarrow \langle Ax, x \rangle \geq \langle Bx, x \rangle, \forall\, x \in \mathcal{X}$. An operator A is said to be *positive semidefinite* if A is a self-adjoint bounded linear operator and $A \succeq 0$. Let $\alpha \in [0, +\infty[$; $\mathcal{P}_\alpha(\mathcal{X})$ denotes the space of positive semidefinite operators A such that $A \succeq \alpha \mathrm{Id}$, i.e., $\mathcal{P}_\alpha(\mathcal{X}) \triangleq \{A \in \mathcal{S}(\mathcal{X}) \,|\, A \succeq \alpha \mathrm{Id}\}$. Given an operator $A \in \mathcal{P}_\alpha$, its *positive square root* $\sqrt{A}$ is the unique operator $\sqrt{A} \in \mathcal{P}_\alpha$ such that $(\sqrt{A})^2 = A$. For every $A \in \mathcal{P}_\alpha(\mathcal{X})$, we define a *semi*-scalar product and a *semi*-norm (a scalar product and a norm if $\alpha > 0$) by $\langle \cdot, \cdot \rangle_A \triangleq \langle A \cdot, \cdot \rangle$ and by $\| \cdot \|_A \triangleq \sqrt{\langle A \cdot, \cdot \rangle}$, respectively. The *domain* of a set-valued operator $A : \mathcal{X} \to 2^{\mathcal{X}}$ is defined by $\mathrm{dom}\, A \triangleq \{x \in \mathcal{X} \,|\, Ax \neq \emptyset\}$, its *graph* by $\mathrm{gra}\, A \triangleq \{(x, u) \in \mathcal{X} \times \mathcal{X} \,|\, u \in Ax\}$, the set of *zeros* by $\mathrm{zer}\, A \triangleq \{x \in \mathcal{X} \,|\, 0 \in Ax\}$, the *range* of A by $\mathrm{ran}\, A \triangleq \{u \in \mathcal{X} \,|\, (\exists x \in \mathcal{X})\ u \in Ax\}$, and the *inverse* of A by $A^{-1} : \mathcal{X} \to 2^{\mathcal{X}} : u \to \{x \in \mathcal{X} \,|\, u \in Ax\}$. We use the notation $\{x^k\}$ as a shorthand for representing the sequence $\{x^k\}_{k=1}^{+\infty}$. We say that a sequence $\{x^k\}$ in $\mathcal{H}$ *converges in the norm* (or *strongly converges*) to a point x in $\mathcal{H}$ if $\|x^k - x\| \to 0$ and say that it *converges weakly* if, for every $u \in \mathcal{H}$, $\langle x^k, u \rangle \to \langle x, u \rangle$. We denote weak convergence by $\xrightarrow{w}$.[3] The space of *absolutely-summable sequences* in $\mathbb{R}$, i.e., the space of sequences $\{x^k\}$ in $\mathbb{R}$ such

[2] The reader may wonder why we do not exclusively consider finite-dimensional spaces, since we are dealing with digital images, which are already discretizations of a continuous space. We choose to consider infinite-dimensional spaces mainly for generalization purposes, since it may be useful to consider these spaces if one wishes to keep the analysis of the algorithms that solve optimization problems independent from the analysis of the discretization of the problem [12].

[3] Strong convergence implies weak convergence to the same limit. In finite-dimensional spaces, weak convergence implies strong convergence.

that $\sum_k |x^k| < \infty$, is denoted by $\ell^1(\mathbb{N})$; the set of summable sequences in $[0, +\infty[$ is denoted by $\ell^1_+(\mathbb{N})$. We denote by $\bigoplus_{j \in \{1,\ldots,N\}} \mathcal{V}_j$ the Hilbert direct sum [11, Example 2.1] of the Hilbert spaces $\mathcal{V}_j$, $j \in \{1, \ldots, N\}$. Given two set-valued operators $A : \mathcal{X} \to 2^{\mathcal{V}}$ and $B : \mathcal{X} \to 2^{\mathcal{V}}$, their *parallel sum* is $A \,\square\, B \triangleq \left(A^{-1} + B^{-1}\right)^{-1}$.

Additionally, we denote by $\mathbb{R}$ the set of real *numbers*, by $\mathbb{R}^n$ the set of real column *vectors* of length n, and by $\mathbb{R}^{m \times n}$ the set of real *matrices* with m rows and n columns. Bold lowercase letters denote vectors and bold uppercase letters denote matrices. $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ denotes the *identity* matrix of size $n \times n$. $\mathbf{1}_n$ denotes a vector of *ones* of size n, and $\mathbf{0}$ denotes a *zero* vector or matrix of appropriate size. $\mathbf{a}^T$ denotes the transpose of a vector $\mathbf{a}$ and $\mathbf{A}^T$ denotes the transpose of a matrix $\mathbf{A}$. $[\mathbf{a}]_i$ denotes the i-th element of a vector $\mathbf{a}$, $[\mathbf{A}]_{:j}$ denotes the j-th column of a matrix $\mathbf{A}$, and $[\mathbf{A}]_{ij}$ denotes the element in the i-th row and j-th column of a matrix $\mathbf{A}$. $\|\mathbf{A}\|_F \triangleq \sqrt{\mathrm{Tr}(\mathbf{A}\mathbf{A}^*)}$ denotes the Frobenius norm of a matrix $\mathbf{A}$.

Finally, let $f : \mathcal{X} \to]-\infty, +\infty]$ be a *function.* Its *domain* is denoted by $\operatorname{dom} f \triangleq \{x \in \mathcal{X} \,|\, f(x) < +\infty\}$ and its *epigraph* by $\operatorname{epi} f \triangleq \{(x, s) \in \mathcal{X} \times \mathbb{R} \,|\, f(x) \leq s\}$. The function f is *lower semi-continuous* if epi f is *closed* in $\mathcal{X} \times \mathbb{R}$, and *convex* if epi f is convex in $\mathcal{X} \times \mathbb{R}$. We use $\Gamma_0(\mathcal{X})$ to denote the class of all lower semi-continuous convex functions f from $\mathcal{X}$ to $]-\infty, +\infty]$ that are *proper*, i.e., such that dom $f \neq \emptyset$. Given two functions $f \in \Gamma_0(\mathcal{X})$ and $g \in \Gamma_0(\mathcal{X})$, their *infimal convolution* is $f \star_{\inf} g : \mathcal{X} \to [-\infty, +\infty] : x \to \inf_{u \in \mathcal{X}} (f(u) + g(x - u))$.

1.1.2. Outline

The structure of this chapter is as follows. Section 1.2 discusses some general characteristics of inverse problems, taking image deconvolution as an example application. Section 1.3 discusses how to formulate imaging problems as regularization ones. Section 1.5 concludes with some comments and extra references.

1.2. Image deblurring as an inverse problem

In this section, we discuss some characteristics of inverse problems. We use the problem of image deblurring as the basis for this analysis.

A grayscale digital image can be represented mathematically as an $m \times n$ matrix, where each element corresponds to a pixel (each pixel is assigned an intensity value). Color or spectral images, which are sets of grayscale images, can be represented as 3-D arrays. However, in this chapter, we consider only grayscale images for ease of exposition. The acquisition process that results in blurred images is usually modeled by a 2-D convolution between a sharp image and the impulse response of a filter. Fig. 1.1 illustrates the situation with an example using the well-known *cameraman* image. The impulse response of the filter is also known as its point-spread function (PSF). This filter is used to model different kinds of image degradation, such as blurs due to motion, atmospheric turbulence, or wrong focus. Additionally, observation noise is also present. It is often assumed to be

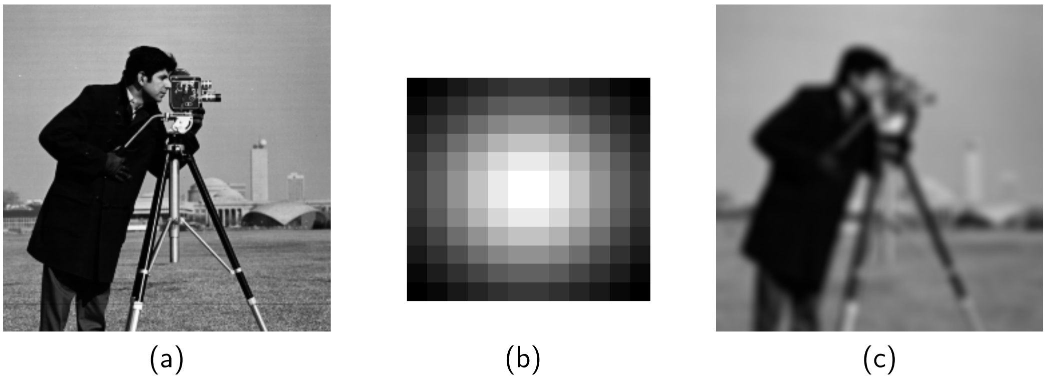


Figure 1.1.: Image deblurring: sharp image (a), filter (b), and blurred image (c), which is the result of the convolution between the sharp image and the filter.

white, Gaussian, and independent from the image. This noise models phenomena with different sources, such as electronic or quantization noise.[4]

In the literature on image processing, it is frequent to model the acquisition process using matrix-vector notation, i.e., the observed (blurred) image, the sharp image and the observation noise are represented not as matrices but as vectors, by ordering these matrices lexicographically. The acquisition process is then modeled as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1.1}$$

where $\mathbf{y} \in \mathbb{R}^{mn}$, $\mathbf{x} \in \mathbb{R}^{mn}$ and $\mathbf{n} \in \mathbb{R}^{mn}$ represent the blurred image, the sharp image, and the noise, respectively, and $\mathbf{H} \in \mathbb{R}^{mn \times mn}$ is a block-Toeplitz-Toeplitz-block (BTTB) matrix such that $\mathbf{Hx}$ represents the convolution of $\mathbf{x}$ with the PSF of the filter. An equivalent formulation would be to represent this convolution as $\mathbf{Xh}$, where $\mathbf{X}$ is a BTTB matrix and $\mathbf{h}$ is a vector representing the filter, but the representation of Eq. (1.1) is more convenient for most applications. Model (1.1) assumes that the blurring effect is uniform throughout the image, i.e., that the degradation is shift-invariant. If this assumption is not verified, the image can be considered to be segmented into different regions, which are usually called *patches* by the image-processing community. In this case, each patch has an associated filter, and model (1.1) is still valid locally, i.e., by considering each patch individually.

The choice of Eq (1.1) to model image deblurring addresses the first point of the framework discussed in Section 1.1. The vector $\mathbf{y}$ corresponds to the observations, and $\mathbf{H}$ and $\mathbf{x}$ correspond to the original data. The direct problem is to find $\mathbf{y}$ given $\mathbf{H}$ and $\mathbf{x}$, whereas the inverse problem is the reverse. Given that blurring is modeled as a convolution, the inverse problem is a *deconvolution* problem, which is usually divided into two categories: *blind* and *non-blind*. In the blind problem, the PSF is unknown, while in the non-blind problem, it is assumed to be known.

[4] For a certain class of images, it might be more convenient to model the noise by a Poisson process.

Problems such as these are usually *ill posed*, in the sense that they do not satisfy any of the three conditions suggested by Hadamard for a problem to be considered well posed, namely: (a) stability, (b) existence, and (c) uniqueness of the solution [13, Ch. 1]. In fact, in regard to point (a), above, the process of computing the inverse solution is often unstable. This means that a small perturbation in some observations might cause large changes in the estimated parameters. In non-blind deconvolution, this statement can be made more precise by studying the singular-value decomposition (SVD) of matrix $\mathbf{H}$, which we do next. In general, the stability condition is the one being violated most often, but the others can also be. On one hand, in regard to point (b), the solution might not even exist, which is true if, e.g., the model does not fit the data exactly. On the other hand, in regard to point (c), even if a solution exists, its uniqueness is not guaranteed. An observation $\mathbf{y}$ could have been generated by an infinite number of pairs of sharp images and filters. For example, in blind deconvolution, all the different $k\mathbf{H}$ and $\frac{\mathbf{x}}{k}$, for $k \in]0, +\infty[$, lead to the same observation $\mathbf{y}$.

For now and until the end of this chapter, unless otherwise noted, we consider that we are dealing with the non-blind deconvolution problem—either $\mathbf{H}$ is known or we have a good estimate of it. A very useful tool to analyze linear systems is the SVD. Matrix $\mathbf{H}$ can be factorized into

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{1.2}$$

where $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{mn\times mn}$ are orthogonal matrices containing the left and right singular vectors, respectively, and $\mathbf{\Sigma} \in \mathbb{R}^{mn\times mn}$ is a diagonal matrix containing the *singular values* $\sigma_1, \cdots, \sigma_{mn} \geq 0$ of $\mathbf{H}$ in non-increasing order. Some of these values might be zero, so assume that there are r positive ones (r corresponds to the rank of the matrix). Denote by $\hat{\mathbf{\Sigma}}$, $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$, respectively, the truncated matrices obtained by discarding the rows and columns with the zero singular values in $\mathbf{\Sigma}$ and the corresponding columns of $\mathbf{U}$ and $\mathbf{V}$. Naturally, $\mathbf{H} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^*$. This can also be written as $\mathbf{H} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$, where $\mathbf{u}_i$ are the columns of $\hat{\mathbf{U}}$ and $\mathbf{v}_i$ are the columns of $\hat{\mathbf{V}}$.

Typically, the smallest singular values of $\mathbf{H}$ are very close to zero. The singular vectors associated with them normally represent high-frequency information [14, Ch. 1]. The condition number of $\mathbf{H}$, given by

$$\text{cond}(\mathbf{H}) \triangleq \frac{\sigma_1}{\sigma_{mn}}, \tag{1.3}$$

is, therefore, very large, or even infinite if $\mathbf{H}$ is not full-rank, i.e., if $r \neq mn$. Problems involving matrices with large condition numbers are called *ill conditioned.*

Imagine that we want to estimate $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{H}$. Ignoring for now the existence of noise, a direct approach is to solve the linear system $\mathbf{y} = \mathbf{H}\mathbf{x}$ for $\mathbf{x}$, which could be done by inverting $\mathbf{H}$, if possible. Since this matrix is not necessarily invertible, the SVD can be used to compute its Moore-Penrose pseudo-inverse [13, Ch. 3]. The solution of the linear system is then given by

$$\hat{\mathbf{x}}_{\text{SVD}} \triangleq \mathbf{H}^\dagger \mathbf{y} \triangleq \hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}^*\mathbf{y} = \sum_{i=1}^{r} \frac{\langle \mathbf{u}_i, \mathbf{y}\rangle}{\sigma_i}\mathbf{v}_i, \tag{1.4}$$

where $\cdot^\dagger$ denotes the pseudo-inverse. However, according to model (1.1), the observation $\mathbf{y}$ is contaminated by noise. Typically, the image $\mathbf{x}$ is mainly associated with the low-frequency components of $\mathbf{y}$ and the noise dominates the high-frequency ones. The solution that we are actually able to find using this approach is $\mathbf{H}^\dagger\mathbf{y} = \mathbf{H}^\dagger\mathbf{H}\mathbf{x} + \mathbf{H}^\dagger\mathbf{n}$, where

$$\mathbf{H}^\dagger\mathbf{n} = \sum_{i=1}^{r} \frac{\langle \mathbf{u}_i, \mathbf{n}\rangle}{\sigma_i}\mathbf{v}_i. \tag{1.5}$$

Eqs. (1.4) and (1.5) involve the reciprocals of the singular values σ_i. Since the ones associated with high-frequency information are very small, their reciprocals are very large. This implies that they have a disproportionate influence on the solution given by this approach and $\mathbf{H}^\dagger\mathbf{y}$ normally yields an image that is extremely noisy.

Under certain assumptions on $\mathbf{H}$, it can be shown that the approach that we have been describing solves a *least-squares* problem [13, Ch. 3], i.e., $\hat{\mathbf{x}}_{\text{SVD}} \in \arg\min_{x\in\mathbb{R}^{mn}}\{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2\}$. This implicitly addresses the second point of the framework discussed in Section 1.1, i.e., the criterion used to evaluate the fitness of the model: we are measuring the ℓ_2 norm of the residual. Other criteria could have been chosen. For example, if the noise was impulsive instead of Gaussian, it would have been more suitable to solve the problem $\arg\min_{x\in\mathbb{R}^{mn}}\{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_1\}$, i.e., to use the ℓ_1 norm instead of the Euclidean norm as criterion.[5]

1.3. Regularization

Due to the issues discussed above regarding the effect of noise, it becomes clear that the pseudo-inverse-based approach cannot be used to solve deconvolution problems. There are a number of alternatives. One of them uses a technique called *truncated SVD* to produce an approximation $\widetilde{\mathbf{H}}$ of $\mathbf{H}$. This approximation is used in place of the original $\mathbf{H}$ in Eq. (1.4) to find a solution to the linear system $\mathbf{y} = \mathbf{H}\mathbf{x}$. $\widetilde{\mathbf{H}}$ is constructed by keeping only the highest singular values of the matrix $\mathbf{H}$, truncating the singular vectors associated with the remaining singular values, i.e., $\widetilde{\mathbf{H}} = \sum_{i=1}^{s} \sigma_i\mathbf{u}_i\mathbf{v}_i^T$ for some choice of $s < r$.[6]

An alternative to stabilize the inversion process that merely attenuates the effect associated with the lowest singular values of $\mathbf{H}$ is a technique called Tikhonov regularization, which consists in solving the problem

$$\underset{\mathbf{x}\in\mathbb{R}^{mn}}{\text{minimize}} \quad \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \mu\|\mathbf{x}\|_2^2, \tag{1.6}$$

where $\mu > 0$ is a regularization parameter that can be adjusted by the user. It can be

[5] In images with noise modeled by a Poisson process, other criteria can be used (see, e.g., Figueiredo and Bioucas-Dias [15]).

[6] This technique finds a low-rank approximation of a given matrix, in the sense that it finds a matrix of rank s that minimizes the Frobenius norm between it and the original matrix.

shown that the solution to Problem (1.6) is given by

$$\hat{\mathbf{x}}_{\text{Tikhonov}} \triangleq \sum_{i=1}^{mn} \alpha_i \frac{\langle \mathbf{u}_i, \mathbf{y} \rangle}{\sigma_i} \mathbf{v}_i, \qquad \alpha_i \triangleq \frac{\sigma_i^2}{\sigma_i^2 + \mu}. \tag{1.7}$$

The terms α_i are called *filter factors*. As before, the observation $\mathbf{y}$ is contaminated by noise, but its effect on $\hat{\mathbf{x}}_{\text{Tikhonov}}$ is attenuated when compared to the solution given in Eq. (1.4). This is because, for $\sigma_i \ll \sqrt{\mu}$, the filter factors are very small (essentially nullifying the components associated with small singular values), while for $\sigma_i \gg \sqrt{\mu}$, they are close to unity.

It is not possible to know beforehand which are the components of $\mathbf{y}$ that should be attenuated. There are principled approaches to determine μ, but sometimes, in practice, Problem (1.6) is solved for different values of μ and the corresponding solutions $\hat{\mathbf{x}}_{\text{Tikhonov}}$ are inspected visually. A very large value of μ leads to a very "smoothed" image, due to the absence of high-frequency information, which is associated with noise but also with edges and other sharp features of the image. In contrast, a very low value of μ leads to a very noisy image. Consequently, the choice of the parameter μ should be made with care, in order to retain the majority of the high-frequency information but without amplifying the noise excessively.

In general, regularization—not necessarily Tikhonov's—implicitly adds extra constraints to our desired solution. For example, the optimization problem

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^{mn}}{\text{minimize}} && \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \\ &\text{subject to} && \|\mathbf{x}\|_2^2 \leq c, \end{aligned} \tag{1.8}$$

with $c > 0$, can be shown to correspond to Problem (1.6) for some μ.[7] In this sense, regularization can be seen as a way of constraining the solution to obey certain conditions. In other words, regularization can be used to add suitable prior information about the solution. Suppose that, as before, we know that our observations follow model (1.1), but now we also know that the desired solution has low energy. Apart from minimizing $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$, it would make sense to also minimize the Euclidean norm of $\mathbf{x}$. This is exactly what is done in Tikhononv regularization. In conclusion, the choice of regularizer corresponds to the third point of the inverse problems framework, i.e., the choice of which additional information about the desired solution to use.

Besides the Euclidean norm, other regularizers acting on $\mathbf{x}$ can be used. The choice of regularizer should reflect prior knowledge about the desired solution. Considering a generic regularizer $\phi \in \Gamma_0(\mathbb{R}^{mn})$, Problem (1.6) can be generalized as

$$\underset{\mathbf{x}\in\mathbb{R}^{mn}}{\text{minimize}} \quad \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \mu\phi(\mathbf{x}). \tag{1.9}$$

Problems such as these are known as *regularization problems*, and the first term is known

[7] See, e.g., Lorenz and Worliczek [16] for a discussion on how these two problems relate to each other.

as the *data-fitting* term.[8] In what follows, we discuss some examples of regularizers and the assumptions associated with them in the context of image deblurring.

The regularizers in Problem (1.9) should reflect *a priori* knowledge about the type of solutions that we seek. For example, as seen before, the term $\mu\|\mathbf{x}\|_2^2$ can be used to reflect the belief that the signal has low energy. A feature of some of the signals that occur in many signal-processing problems is *sparsity.* If a signal is represented by a vector, it is said to be sparse if most of its coefficients are zero. One way to use this fact when formulating a problem such as Problem (1.9) is to use the ℓ_0 pseudo-norm of $\mathbf{x}$ as a regularizer, which counts the number of non-zero entries of $\mathbf{x}$. The resulting optimization problem is

$$\underset{\mathbf{x}\in\mathbb{R}^{mn}}{\text{minimize}} \quad \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \mu\|\mathbf{x}\|_0. \tag{1.10}$$

The use of the ℓ_0 pseudo-norm leads to a problem that is NP-hard [17]. Under certain assumptions on $\mathbf{H}$, it has been shown that the solution to this problem coincides, with high probability, with the solution of a problem that is much easier to solve[9] (see, e.g., Candès [19] and references within). This problem is

$$\underset{\mathbf{x}\in\mathbb{R}^{mn}}{\text{minimize}} \quad \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \mu\|\mathbf{x}\|_1. \tag{1.11}$$

Unlike the ℓ_0 pseudo-norm, the ℓ_1 norm is convex, a fact that can be taken advantage of when solving optimization problems (these problems are discussed in Section 2.1).

In imaging problems, not all images are sparse, but the majority of them are still highly structured. Natural images, for example, are “piecewise smooth”, i.e., they can be broken into distinct regions where the intensities of the pixels are almost constant. This means that, in domains such as the derivative and wavelet ones, the signal is approximately sparse, in the sense that the majority of the entries of the vector representing the signal are not zero, but small. An optimization problem that reflects the use of this *a priori* knowledge is

$$\underset{\mathbf{x}\in\mathbb{R}^{mn}}{\text{minimize}} \quad \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \mu\|\mathbf{D}\mathbf{x}\|_1, \tag{1.12}$$

[8] Another way to look at the discussed minimization problems is through a Bayesian framework. In fact, image deconvolution can be expressed as a solution to a maximum-*a-posteriori* (MAP) problem. Under a Bayesian framework, all parameters are viewed as unknown stochastic variables, which have associated probability distributions reflecting prior beliefs. These distributions depend on hyperparameters, which we denote here by Ω and assume to be known. Given observations $\mathbf{y}$ and $\mathbf{H}$, and parameter $\mathbf{x}$, we can write, using Bayes' law,

$$p(\mathbf{x}|\mathbf{H},\mathbf{y};\Omega) \propto p(\mathbf{y}|\mathbf{x},\mathbf{H};\Omega)p(\mathbf{x};\Omega),$$

where $p(\mathbf{x}|\mathbf{H},\mathbf{y};\Omega)$ is the posterior probability and $p(\mathbf{x}|\Omega)$ is the prior on $\mathbf{x}$. By making

$$p(\mathbf{y}|\mathbf{x},\mathbf{H};\Omega) \propto e^{-\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2}$$

and

$$p(\mathbf{x};\Omega) \propto e^{-\mu\phi(\mathbf{x})},$$

it is clear that maximizing $p(\mathbf{x}|\mathbf{H},\mathbf{y};\Omega)$ to find the most probable $\mathbf{x}$ is equivalent to solving Problem (1.9).

[9] Loosely speaking, the rows of $\mathbf{H}$ are assumed to be incoherent (see, e.g., Candès and Plan [18] for a precise definition of this term). This assumption is not valid for typical convolution operators, but it may still be useful in practice to solve Problem (1.11) instead of Problem (1.10).

where $\mathbf{D} \in \mathbb{R}^{k\times mn}$ is a linear operator with respect to which $\mathbf{x}$ has an approximately sparse representation. Other examples of linear operators that could be of interest are matrices whose rows are the entries of dictionaries learned from the data itself [20, 21], or operators with respect to which $\mathbf{x}$ has a redundant representation [22].

Another commonly used regularizer that encodes the fact that images are "piecewise smooth" is total variation (TV) [23]. Two popular formulations of this regularizer are the often called anisotropic TV and isotropic TV. They are given by

$$\phi_{TV}(\mathbf{x}) \triangleq \sum_{i=1}^{mn} \left\| \begin{bmatrix} [\mathbf{D}_h\mathbf{x}]_i \\ [\mathbf{D}_v\mathbf{x}]_i \end{bmatrix} \right\|_p, \tag{1.13}$$

where $\mathbf{D}_h$ and $\mathbf{D}_v$ are such that the products by these matrices compute, respectively, the horizontal and vertical first-order differences of a discrete image. If $p = 1$, we recover the anisotropic TV case, whereas if $p = 2$, we recover the isotropic one. Note that $\phi_{TV}(\mathbf{x}) \in \Gamma_0(\mathbb{R}^{mn})$.

Apart from sparsity, it can also be useful to impose some other constraints on our desired solution, such as requiring it to be nonnegative (which is the case of any image) or its domain to be bounded. These constraints can be added to the minimization problem as proper constraints or through the use of the *indicator function* of a set $C \in \mathbb{R}^{mn}$, which is defined as

$$\delta_C(\mathbf{x}) \triangleq \begin{cases} 0 & \text{if } \mathbf{x} \in C, \\ +\infty & \text{otherwise.} \end{cases} \tag{1.14}$$

When C is convex, closed and non-empty, $\delta_C(\mathbf{x}) \in \Gamma_0(\mathbb{R}^{mn})$. An example of its use as a regularizer is in the problem

$$\underset{\mathbf{x}\in\mathbb{R}^{mn}}{\text{minimize}} \quad \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \mu\|\mathbf{x}\|_1 + \delta_{[0,1]^{mn}}(\mathbf{x}), \tag{1.15}$$

which imposes that the intensity of each pixel be in $[0, 1]$. Unlike the solution to Problem (1.6), which has a closed form, Problems (1.12) and (1.15) are solved through iterative algorithms. These algorithms are discussed in Chapter 2.

Blind deconvolution problems (i.e., problems where the PSF is unknown) can also be tackled as optimization problems. There is one extra variable to estimate (the filter's PSF), and it can be useful to impose certain conditions on it (e.g., low energy). We consider a regularizer on the image, $\phi_1 \in \Gamma_0(\mathbb{R}^{mn})$, as well as one on the filter, $\phi_2 \in \Gamma_0(\mathbb{R}^{mn})$,

$$\underset{\mathbf{x}\in\mathbb{R}^{mn},\,\mathbf{h}\in\mathbb{R}^{mn}}{\text{minimize}} \quad \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \mu_1\phi_1(\mathbf{x}) + \mu_2\phi_2(\mathbf{h}). \tag{1.16}$$

This problem is much harder to solve than non-blind deconvolution for two reasons: (a) we need to estimate both $\mathbf{x}$ and $\mathbf{h}$, and (b) the data-fitting term is not convex anymore. An approach that is used in practice is to alternately solve for $\mathbf{x}$ (fixing $\mathbf{h}$) and then for $\mathbf{h}$ (fixing $\mathbf{x}$), the first of these steps effectively solving a non-blind deconvolution problem of the type of Problem (1.9).

1.4. Remote-sensing hyperspectral imaging

In this section, we give a brief overview on remote-sensing hyperspectral imaging, and, in particular, on the spectral-unmixing problem. This problem is not analyzed in subsequent chapters, but algorithms that address it are used as building blocks of some of the algorithms discussed in Parts II and III, so we briefly discuss them here.

Hyperspectral imaging (often called *imaging spectroscopy*) is an extension of color imaging. The latter is concerned with images that only have three channels—these correspond to the blue, green and red wavelengths of visible light—whereas the former is concerned with images with many more channels. Each channel (which is also called *spectral band*) captures a different frequency range along the EM spectrum. Hyperspectral images find applications in surveillance, defense, agriculture, mineralogy, geology, astronomy, environmental science, and the food and pharmaceutical industries. In this work, we consider the images used in the remote-sensing field. These images cover the visible, near-infrared and short-wavelength infrared spectral ranges, i.e., they cover the wavelength range from 400 nm to 2500 nm.

In Earth remote-sensing imaging, images representing the reflectance of a given scene are routinely acquired from air- or space-borne sensors. These images are typically *panchromatic* (i.e., single-channel) and/or *multispectral* images, but, increasingly, one can also find sensors acquiring *hyperspectral* images. An example of such a sensor is the Hyperion Imaging Spectrometer [on board the Earth Observing-1 Mission (EO-1) satellite], which captures images with circa 200 channels, each corresponding to a band of the EM spectrum covering a range of approximately 10 nm, with a spatial resolution of 30 m. An example of a more commonly found satellite is IKONOS, whose instruments acquire multispectral images (MSIs) with four channels (blue, green, red, and near-infrared) and with a spatial resolution of 3.2 m; they also acquire panchromatic images (PANs) with a spatial resolution of 0.82 m. PANs usually cover the visible and the near-infrared spectral ranges, but do not cover higher wavelength ranges. In contrast, both hyperspectral images (HSIs) and MSIs usually cover a larger spectral range. Typically, these images also have a higher spectral resolution than PANs, but a lower spatial resolution. In particular, HSIs have a still higher spectral resolution than MSIs, but typically have a lower spatial resolution than either MSIs or PANs. The differences in resolution are due to the design of the instruments that acquire these images. In general, it is not easy to build sensors that produce images with both a high spectral and a high spatial resolution.[10]

HSIs, due to their large number of channels, contain more spectral information than MSIs, a fact that makes these images very useful in applications such as object detection, and material classification and identification. While, in the past, hyperspectral sensors were not common, they are becoming more and more popular. For example, a number of missions that have the goal of launching satellites with hyperspectral sensors are currently under development worldwide: the German Environmental Mapping and Analysis Program (EnMAP) [24] is preparing the launch of the Polar Satellite Launch Vehicle,

[10] The definition of what constitutes a multispectral or hypersepctral image is not exact: for example, the EO-1 satellite captures HSIs and MSIs with the same spatial resolution.

which includes a hyperspectral sensor, in 2018;[11] the Italian mission PRecursore IperSpettrale della Missione Applicativa (PRISMA) [25] is planning the launch of a satellite containing both hyperspectral and panchromatic capturing devices in 2018;[12] the Indian satellite GEO Imaging Satellite (GISAT) 1 and the ones from mission Cartosat-3 are due to launch in 2017-2020—the three will carry hyperspectral sensors and the latter will also carry panchromatic and multispectral sensors—;[13] NASA's Hyperspectral InfraRed Imager (HyspIRI) [26] will include a hyperspectral imager and a thermal infrared scanner;[14] and the Japanese Hyperspectral Imager SUIte (HISUI) will include hyperspectral and multispectral sensors [27].[15]

The relatively low spatial resolution of HSIs causes several problems, and it is of interest to produce images with higher resolutions [28]. One way to accomplish this is by means of superresolution algorithms. They can be seen as a processing step to be performed in the first stage of the pipeline of some applications, such as object detection or spectral unmixing.

Multispectral and hyperspectral images can be thought of as *data cubes* (3-D arrays). Two of the dimensions of the cube are associated with physical space, whereas the other one is associated with wavelength. Another representation that is perhaps more commonly found in the literature is the one that considers HSIs and MSIs to be matrices. Each row corresponds to the lexicographical ordering of the image associated with one channel. Using this notation, HSIs can be represented by a matrix $\mathbf{Y}_h \in \mathbb{R}^{L_h \times n_h}$, with L_h channels and spatial dimension n_h, and MSIs can be represented by a matrix $\mathbf{Y}_m \in \mathbb{R}^{L_m \times n_m}$, with L_m channels and spatial dimension n_m. Typically, $L_m < L_h$ and $n_m > n_h$.

1.4.1. Spectral unmixing

A problem caused by the relatively low resolution of hyperspectral images, among other reasons, is spectral mixing. Consider, for example, a hyperspectral sensor aboard a satellite acquiring images above a forest. If one wishes to identify the species comprising this forest, one may look at the reflectance intensities along the EM of each pixel and compare them to a database of spectral signatures (i.e., a database of reflectance profiles as a function of wavelength) of trees and other materials. Each material (which in this context is usually called *endmember*) has its own spectral signature. In general, a pixel of a HSI does not correspond to just a single material, but to a region with several of them. Consequently, the reflectance profile of a single pixel will correspond to a mixture of endmember spectral signatures. In order to disentangle this mixture, spectral

[11] See `http://www.enmap.org/` for details.

[12] See `http://www.asi.it/en/flash/observing-earth/prisma`, `http://space.skyrocket.de/doc_sdat/prisma_asi.htm`, and `https://directory.eoportal.org/web/eoportal/satellite-missions/p/prisma-hyperspectral` for details.

[13] See`http://space.skyrocket.de/doc_sdat/gisat-1.htm` and `http://space.skyrocket.de/doc_sdat/cartosat-3.htm` for details.

[14] See `https://hyspiri.jpl.nasa.gov/` for details.

[15] See `http://www.jspacesystems.or.jp/en_project_hisui/` for details.

unmixing techniques can be used. They produce a set of spectral profiles, one for each endmember, and a corresponding set of *abundances*, or percentages of occupation, for each endmember, in each pixel. Spectral unmixing is an ill-posed problem.

The spectral mixture can be considered to be linear,[16] in the sense that the reflectance profile of a given pixel corresponds to a weighted sum of the spectral signatures of the endmembers present in the region associated with this pixel. Linear spectral mixing can be modeled pixel-wise as

$$[\mathbf{Y}_h]_{:j} = \mathbf{U}\mathbf{a}_j + \mathbf{n}_j, \quad j = \{1, \cdots, n_h\}, \tag{1.17}$$

where $[\mathbf{Y}_h]_{:j}$ is a vector containing in its elements the reflectance profile of pixel j of a given hyperspectral image $\mathbf{Y}_h$, $\mathbf{U} \in \mathbb{R}^{L_h \times P}$ is a matrix whose columns contain the spectral signature of the various endmembers, P is the number of endmembers present in the scene, $\mathbf{a}_j \in \mathbb{R}^P$ is a vector containing in its elements the abundance fractions of the endmembers present in pixel j, and $\mathbf{n}_j \in \mathbb{R}^{L_h}$ is a vector representing noise and/or modeling errors. The matrix $\mathbf{U}$ is assumed to be known if it corresponds to a database (or dictionary) of spectral signatures acquired in laboratory of different materials, or is estimated from the data itself.

The abundances obey two conditions: they are nonnegative and they sum to one, i.e.,

$$\begin{cases} [\mathbf{a}_j]_i \geq 0, \quad i = \{1, \cdots, P\}, \\ \mathbf{1}_P^T \mathbf{a}_j = 1. \end{cases} \tag{1.18}$$

As a consequence, the vectors of abundances $\mathbf{a}_j$, $j = \{1, \cdots, n_h\}$, are contained in the simplex of dimension $(P-1)$ if the columns of $\mathbf{U}$ are affinely independent.[17] Each vertex of the simplex corresponds to the spectral signature of one endmember. This fact has been taken advantage of in the design of many of the unmixing algorithms found in the literature.

Unmixing algorithms can be classified as being *pure-pixel*-based or not. Pure-pixel-based algorithms are built under the assumption that the scene represented in the hyperspectral image has regions where only one material is observed. Additionally, it is assumed that such regions exist for every material. These regions should be represented by one (or more) pixels in the image, which are called pure pixels. In other words, each vertex of the $(P-1)$-simplex is present in the columns of the matrix $\mathbf{Y}_h$. Although such assumptions may be too strong for many real-world images, pure-pixel-based algorithms tend to be computationally fast and conceptually intuitive. An example of a pure pixel-based algorithm is vertex component analysis (VCA) [31].

Other algorithms that are used to spectrally unmix HSIs do not rely on the existence of pure pixels. Some of them exploit the fact that each pixel contains only a small number of materials, i.e., that $\mathbf{a}_j$ is sparse, $j = \{1, \cdots, n_h\}$. They are based on the solution to an

[16] This is true only under certain assumptions [29], but linearity is an acceptable assumption for many practical scenarios. One could also consider nonlinear models [30].

[17] I.e., if $[\mathbf{U}]_{:1} - [\mathbf{U}]_{:0}, [\mathbf{U}]_{:2} - [\mathbf{U}]_{:0}, \cdots, [\mathbf{U}]_{:P} - [\mathbf{U}]_{:0}$ are linearly independent.

optimization problem where sparsity is imposed on $\mathbf{a}_j$ via a sparsity-inducing regularizer. A possible formulation, similar to the one of Problem (1.11), is given, pixel-wise, by

$$\begin{aligned} \underset{\mathbf{a}_j \in \mathbb{R}^P}{\text{minimize}} \quad & \|[\mathbf{Y}_h]_{:j} - \mathbf{U}\mathbf{a}_j\|_2^2 + \mu\|\mathbf{a}_j\|_1 \\ \text{subject to} \quad & [\mathbf{a}_j]_i \geq 0, \quad i = \{1, \cdots, P\}. \end{aligned} \tag{1.19}$$

In these algorithms, the condition $\mathbf{1}_P^T \mathbf{a}_j = 1$ is usually not enforced (see Bioucas-Dias et al. [29, Section VI] for an explanation).

1.5. Comments and references

The characterization of inverse problems given here was inspired by Hansen et al. [14], Press et al. [32], and Aster et al. [13].

For more details on the use of sparsity in imaging problems, see Bach [33]. For more details on algorithmic approaches to the problem of image deblurring, see Campisi and Egiazarian [34] and Rajagopalan and Chellappa [35]. For more details on the problem of image superresolution, see Milanfar [36].

For an introduction to the topic of remote-sensing imaging, see Shaw and Burke [28], and Bioucas-Dias et al. [37]. For more details on spectral unmixing, see Dobigeon et al. [30] and Bioucas-Dias et al. [29].

2. Algorithms used to solve imaging problems

This chapter addresses the last point of the framework discussed in Section 1.1, i.e., the algorithms used to solve imaging optimization problems. These problems are considered to be hard to solve, and normally it is not possible to use a general-purpose optimization algorithm to solve them due, essentially, to two reasons. First, these algorithms are often very slow when dealing with large-scale problems, as normally happens in image processing. Second, in order to be able to take advantage of the aforementioned sparsity of natural images when represented in appropriate domains, imaging optimization problems frequently involve regularizers that are non-differentiable. For these two reasons, it is of great interest to develop specific algorithms to solve imaging problems. In this chapter, we present some of these algorithms. We start by introducing some notions used in the study of convex optimization problems. We then describe some basic algorithms that are used as building blocks for more complex ones, which we also present next. Finally, we discuss these algorithms in the context of monotone-operator and fixed-point theory.

Contents

2.1. Introduction

In this work, all the optimization problems under consideration are convex. A distinctive feature of this class of problems is that, if a local minimum exists, it must be a global minimum. For this reason, convex problems are usually much easier to solve than non-convex ones. We assume that all problems under consideration have at least one minimizer.

The algorithms used to solve imaging problems are usually iterative, since it is impractical or even impossible to find a closed-form solution to these problems. Different criteria are used to decide when to stop the algorithms, e.g., if the number of iterations is larger than a predetermined value or if the ℓ_2 norm between the solutions of two consecutive iterations is lower than a given threshold. When devising iterative algorithms, it is usually important to study what are the conditions under which they converge, and the rate at which they do so. For many algorithms, this study can be done in a systematic way by formulating them as fixed-point iterations of appropriate operators. This topic is addressed in Sections 2.3 and 2.5.

We now introduce some basic concepts used in convex analysis. Although all functions under consideration are convex, some of them are not smooth, i.e., they are non-differentiable (in the sense of Fréchet [11, Definition 2.45]). The notions of subgradient and subdifferential of a convex function (in the sense of Moreau and Rockafellar [38, Chapter 23]) are useful when dealing with them. A vector $\mathbf{p} \in \mathbb{R}^n$ is said to be a *subgradient* of a function $g \in \Gamma_0(\mathbb{R}^n)$ at a point $\mathbf{x} \in \mathbb{R}^n$ if

$$g(\mathbf{y}) \geq g(\mathbf{x}) + \langle \mathbf{p}, \mathbf{y} - \mathbf{x} \rangle, \quad \forall\, \mathbf{y} \in \mathbb{R}^n. \tag{2.1}$$

The set of all subgradients of g at $\mathbf{x}$ is called the *subdifferential of g at* $\mathbf{x}$ and is denoted by $\partial g(\mathbf{x})$. The set-valued operator $\partial g : \mathbb{R}^n \to 2^{\mathbb{R}^n} : \mathbf{x} \to \partial g(\mathbf{x})$ is called the *subdifferential* of g. For a differentiable function $f \in \Gamma_0(\mathbb{R}^n)$, the subdifferential at $\mathbf{x}$ is a singleton, i.e., $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$. The subdifferential operator is critical to our interests. We recall Fermat's rule [11, Theorem 16.2]: $\mathbf{x}$ is a minimum of a proper convex function if and only if $\mathbf{0} \in \partial g(\mathbf{x})$.

We say that an operator A is *Lipschitz continuous* with constant $L > 0$ if

$$\|u - v\| \leq L\|x - y\|, \quad \forall\,(x, u) \in \text{gra } A, \quad \forall\,(y, v) \in \text{gra } A. \tag{2.2}$$

For example, the data-fitting term of the objective function of Problem (1.9), $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$, has a Lipschitz-continuous gradient with constant $\beta = 2\|\mathbf{H}\|^2$. When $L = 1$ in (2.2), the operator A is said to be *nonexpansive*; when $L < 1$, it is said to be *contractive.*

2.1.1. Outline

In Chapter 1, we discussed how the objective functions of convex problems in the context of imaging usually involve data-fitting terms and one or more regularizers. We can consider these problems as particular cases of the problem

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}) + \sum_{j=1}^{N} h_j(\mathbf{D}_j \mathbf{x}), \tag{2.3}$$

where $f \in \Gamma_0(\mathbb{R}^n)$ is differentiable, but $g \in \Gamma_0(\mathbb{R}^n)$ and the N functions $h_j \in \Gamma_0(\mathbb{R}^{m_j})$ may not be, and $\mathbf{D}_j \in \mathbb{R}^{m_j \times n}$, $j = \{1, \ldots, N\}$.

This chapter lists and briefly discusses a number of algorithms from the literature that can be used to solve Problem (2.3) and some of its particular cases. The choice of which algorithms to list was very practical: either they are explicitly discussed in Parts II and III, or, if not, they are used as building blocks for the ones that are. We start by analyzing the simplest ones. More specifically, Section 2.2 discusses algorithms used to solve problems with smooth objective functions—i.e., problems of the form of Problem (2.3) when $g = 0$ and $h_j = 0$, $j \in \{1, \ldots, N\}$—and algorithms used to solve problems with nonsmooth objective functions—i.e., problems of the form of Problem (2.3) when $f = 0$ and $h_j = 0$, $j \in \{1, \ldots, N\}$. Section 2.3 presents the forward–backward method, which can be used to solve Problem (2.3) when $h_j = 0$, $j \in \{1, \ldots, N\}$, and the Douglas–Rachford method, which can be used to solve Problem (2.3) when $f = 0$ and $h_j = 0$, $j \in \{2, \ldots, N\}$. That section also introduces some notions on Fenchel's and Lagrange's duality, and discusses the alternating-direction method of multipliers. Finally, it presents an algorithm that is able to efficiently solve Problem (2.3) itself. Section 2.4 analyzes a class of algorithms known as semismooth Newton methods, as well as their connection to active-set methods. Section 2.5 gives a brief overview on monotone-operator theory, and on how it relates to optimization problems and algorithms. Section 2.6 concludes with some comments and extra references.

We consider that the algorithms under discussion in this chapter are being used to solve digital-imaging problems. Consequently, in Sections 2.2-2.4, we limit their analysis to finite-dimensional real Hilbert spaces. However, these algorithms could also be presented in the more general context of infinite-dimensional spaces. Their study from the point of view of monotone operators is given in infinite-dimensional spaces (Section 2.5).

2.2. Smooth and nonsmooth minimization

In this section, we consider the minimization of two types of convex functions: differentiable and non-differentiable ones.

Imagine that we want to find a minimizer of a function $f \in \Gamma_0(\mathbb{R}^n)$ that has a Lipschitz-continuous gradient,

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}). \tag{2.4}$$

A standard method to find this minimizer is the iterative algorithm known as *gradient descent*,

Algorithm 1: Gradient-descent method with a fixed step-size parameter.

```
Choose $\mathbf{x}^0 \in \mathbb{R}^n$, $\tau > 0$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - \tau \nabla f(\mathbf{x}^k)$;
    $k \leftarrow k + 1$;
end
```

where k denotes the iteration number and $\tau > 0$ is a fixed step-size parameter. The gradient-descent method is a particular case of a *line-search* method,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda^k \mathbf{p}^k, \quad \mathbf{x}^k \in \mathbb{R}^n, \tag{2.5}$$

where $\lambda^k > 0$ is the *step-size* parameter and $\mathbf{p}^k$ is a vector indicating a *search direction.* Usually, $\mathbf{p}^k$ is required to be a descent direction, i.e., $\langle \mathbf{p}^k, \nabla f(\mathbf{x}^k)\rangle < 0$, since this property guarantees that f is reduced along this direction [39, Chapter 2]. By making $\mathbf{p}^k = -\nabla f(\mathbf{x}^k)$, we recover the gradient-descent method. This algorithm is a *first-order* method, since only the *first* derivative of f is assumed to be known.

Although many problems in imaging are typically large-scale, the gradient-descent method and, in particular, its extensions, can be used to efficiently tackle them. The extensions of this method use a "divide-and-conquer" strategy by splitting the initial problem into a series of simpler subproblems (see Section 2.3 for more details). Each iteration of these methods is usually simple to compute. When such methods are run on a modern computer, they are able to find a medium-to-low-accuracy solution in a reasonable amount of time for many problems of interest. The solution is generally good enough for most practical purposes, since the human eye is relatively insensitive to small differences in the intensity values of an image.

The convergence rate of the gradient-descent method can often be improved through the use of *relaxed* and/or *inertial* steps, or with the use of *second-order* information if f is twice-differentiable.[1] We introduce relaxed steps and inertial steps in Section 2.3, and here we briefly discuss second-order line-search methods. We generalize them to variable-metric methods in Section 2.3 as well.

In second-order line-search methods, the search direction is given by $\mathbf{p}^k = -[\mathbf{B}^k]^{-1}\nabla f(\mathbf{x}^k)$, where $\mathbf{B}^k \in \mathbb{R}^{n\times n}$ is an invertible matrix that obeys certain conditions, such as being positive definite (in order for $\mathbf{p}^k$ to be guaranteed to be a descent direction). If $\mathbf{B}^k = \nabla^2 f(\mathbf{x}^k)$, we recover the classical *Newton* method, which uses exact knowledge of the Hessian $\nabla^2 f$ of the smooth function. The Hessian may not always be positive definite, and in that case, $\mathbf{B}^k$ could be modified by, e.g., adding a multiple of the identity such that the resulting matrix is sufficiently positive definite, or truncating/replacing its nonpositive eigenvalues [39, Chapter 3]. Furthermore, computing the Hessian may not be convenient, and, in that case, an approximation can be used, $\mathbf{B}^k \approx \nabla^2 f(\mathbf{x}^k)$. Methods that use approximations are sometimes called *quasi-Newton.* Under certain assumptions, the local convergence rate of second-order line-search methods is superlinear or even quadratic, whereas the convergence rate of the gradient-descent method is linear.[2]

[1] In this work, the classification of the convergence rates of algorithms are as described in Nesterov [40, Section 1.2.4] and Nemirovksi [41, Section 1.3.3]. There, they are classified as *sublinear*, *linear*, *superlinear*, or *quadratic*. In theory, an algorithm with a quadratic convergence rate is faster than one with a superlinear rate, which, in turn, is faster than one with a linear rate, etc. In practice, this only holds if the computational complexity per iteration of the algorithms is comparable. It is common to study the convergence rate of second-order line-search methods both locally and globally. By *locally*, we mean that we study these methods while assuming that the iterates generated by them are close enough to the solution, so that the Hessian of the objective function at that point is positive definite. By *globally*, we mean that we study these methods when this assumption is not made.

[2] Usually, under the assumption that f is *strongly convex*, i.e., that $f - \frac{m}{2}\langle \mathbf{x}, \mathbf{x}\rangle$ is convex, for some

Another line-search method of interest is the well-known *conjugate-gradient* method, which is a first-order method. It can be used to minimize a convex quadratic function $f_Q : \mathbb{R}^n \rightarrow]-\infty, +\infty] : \mathbf{x} \rightarrow \frac{1}{2}\langle \mathbf{x}, \mathbf{A}\mathbf{x}\rangle - \langle \mathbf{b}, \mathbf{x}\rangle + c$, where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a positive-definite matrix, $\mathbf{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$. The method attains the optimum in at most n steps. It is faster than gradient descent, and does not require any storage of matrices [39, Chapter 5]. It can also be used to solve the linear equation $\mathbf{A}\mathbf{x} = \mathbf{b}$, since solving this system is equivalent to minimizing f_Q. Note, however, that the conjugate-gradient is not used exclusively to find minimizers of quadratic functions. In fact, nonlinear versions of this method can be used to minimize general nonlinear functions.

Consider now that we want to find the minimizer of a convex nonsmooth function $g \in \Gamma_0(\mathbb{R}^n)$, such as the ℓ_1 regularizer mentioned in Section 1.3,

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad g(\mathbf{x}). \tag{2.6}$$

The gradient-descent method cannot be used, since g is not differentiable everywhere. One alternative is to minimize a smooth surrogate of this function. The *Moreau envelope* is such a surrogate. It is defined by

$$^{\tau}g(\mathbf{x}) \triangleq \inf_{\mathbf{u}\in\mathbb{R}^n} \left\{ g(\mathbf{u}) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{u}\|^2 \right\}, \tag{2.7}$$

where $\tau > 0$. $^{\tau}g$ is continuously differentiable, even if g is not. Both functions share the same minimizers [11, Proposition 12.9(iii)], and therefore the problems of minimizing g and $^{\tau}g$ are equivalent.[3]

Before continuing, we introduce the *proximal operator* of g,

$$\text{prox}_{\tau g}(\mathbf{x}) \triangleq \arg\min_{\mathbf{u}\in\mathbb{R}^n} \left\{ g(\mathbf{u}) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{u}\|^2 \right\}, \tag{2.8}$$

which is simply the point that achieves the infimum of Eq. (2.7) (this point is unique, since $\mathbf{u} \rightarrow g(\mathbf{u}) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{u}\|^2$ is strictly convex). The proximal operator of g relative to

$m > 0$. Roughly speaking, for the problems under consideration in this work, first-order methods will converge sublinearly. The use of relaxed and/or inertial steps may improve the convergence rate, but it will remain sublinear. If the objective function being minimized is strongly convex, the convergence rate may be linear. Under the same assumption on the objective function, the use of second-order information may further improve the local convergence rate to superlinear (if a certain approximation to the Hessian is used [42]) or to quadratic (if the Hessian itself is used).

[3] We give here an intuitive explanation, inspired by signal processing, on why the Moreau envelope is a smooth surrogate of g, since many parallelisms can be drawn between harmonic and convex analysis. The Moreau envelope of g is, in fact,

$$^{\tau}g(\mathbf{x}) = \tau g(\mathbf{x}) \star_{\inf} \frac{1}{2}\|\mathbf{x}\|^2.$$

We note that $\|\cdot\|^2$ is a smooth function and, in a sense, we are smoothing the sharp discontinuities that might exist in g. For more on the analogy between the Legendre–Fenchel conjugate and the infimal convolution, and between the Fourier transform and the integral convolution, see, e.g., Komodakis and Pesquet [43]. The notion of conjugate is introduced in Section 2.3.

the norm $\|\cdot\|_{\mathbf{U}}^2$, where $\mathbf{U} \in \mathbb{R}^{n\times n}$ and $\mathbf{U} \succ 0$, is

$$\operatorname{prox}_{\tau g}^{\mathbf{U}}(\mathbf{x}) \triangleq \underset{\mathbf{u}\in\mathbb{R}^n}{\arg\min} \left\{ g(\mathbf{u}) + \frac{1}{2\tau}\|\mathbf{x}-\mathbf{u}\|_{\mathbf{U}}^2 \right\}. \tag{2.9}$$

The Moreau envelope is continuously differentiable, with gradient [11, Proposition 12.29]

$$\nabla\left[{}^{\tau}g(\mathbf{x})\right] = \frac{1}{\tau}\left[\mathbf{x} - \operatorname{prox}_{\tau g}(\mathbf{x})\right], \quad \forall \mathbf{x} \in \mathbb{R}^n. \tag{2.10}$$

The latter equation can be rewritten as

$$\operatorname{prox}_{\tau g}(\mathbf{x}) = \mathbf{x} - \tau\nabla\left[{}^{\tau}g(\mathbf{x})\right], \quad \forall \mathbf{x} \in \mathbb{R}^n, \tag{2.11}$$

and, by noting the similarities between Eq. (2.11) and Line 4 of Algorithm 1, one can devise a scheme similar to Algorithm 1 to solve problems involving nonsmooth functions, the *proximal-point* method,

Algorithm 2: Proximal-point method with a fixed step-size parameter.

```
Choose $\mathbf{x}^0 \in \mathbb{R}^n$, $\tau > 0$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    $\mathbf{x}^{k+1} \leftarrow \operatorname{prox}_{\tau g}(\mathbf{x}^k) = \underset{\mathbf{u}\in\mathbb{R}^{\ltimes}}{\arg\min}\left\{g(\mathbf{u}) + \frac{1}{2\tau}\|\mathbf{x}^k - \mathbf{u}\|^2\right\}$;
    $k \leftarrow k + 1$;
end
```

We now list the proximal operators of some functions that are used in Parts II and III.[4] When $g(\mathbf{x}) = \|\mathbf{x}\|_1$, $\operatorname{prox}_{\tau g}(\mathbf{x})$ can be evaluated component-wise. The proximal operator for each element reduces to the so-called *soft-thresholding* operator [45], which is given by

$$[\operatorname{prox}_{\tau\|\cdot\|_1}(\mathbf{x})]_i = \max\{|[\mathbf{x}]_i| - \tau, 0\}\operatorname{sgn}([\mathbf{x}]_i), \quad \forall i \in \{1, \cdots, n\}. \tag{2.12}$$

Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be vectors in $\mathbb{R}^n$, let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, and let $g(\mathbf{x}) = \sum_{i=1}^n \|([\mathbf{x}_1]_i, [\mathbf{x}_2]_i)\|_2$. The proximal operator of g can be evaluated component-wise. For each i, it reduces to the so-called *vector soft-thresholding* operator [46], which is given by

$$[\operatorname{prox}_{\tau g}(\mathbf{x})]_i = \max\left\{\|([\mathbf{x_1}]_i, [\mathbf{x_2}]_i)\|_2 - \tau, 0\right\} \frac{([\mathbf{x_1}]_i, [\mathbf{x_2}]_i)}{\|([\mathbf{x_1}]_i, [\mathbf{x_2}]_i)\|_2}, \quad \forall i \in \{1, \cdots, n\}. \tag{2.13}$$

We follow the convention that $0/\|\mathbf{0}\|_2 = 0$.

Finally, the proximal operator of the indicator function [cf. Eq. (1.14)] when the set $C \in \mathbb{R}^n$ is convex, closed and non-empty is an Euclidean projection onto C, which we denote by

$$P_C(\mathbf{x}) \triangleq \underset{\mathbf{u}\in C}{\arg\min}\|\mathbf{x}-\mathbf{u}\|_2^2. \tag{2.14}$$

[4] See, e.g., Combettes and Pesquet [44] for the proximal operators of more functions.

2.3. Splitting methods

The objective functions of many problems of interest, such as the one of Problem (1.9), can be formulated as sums of two convex functions: one smooth, $f \in \Gamma_0(\mathbb{R}^n)$, and another one that needs not be, $g \in \Gamma_0(\mathbb{R}^n)$. The resulting problem is

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}). \tag{2.15}$$

Such a problem could be solved by using the proximal-point method, since $f + g$ is itself a nonsmooth convex function. This would require one to compute $\operatorname{prox}_{(f+g)}$, which is frequently hard to do. Sometimes, it is useful to consider *splitting methods*, which are methods that take advantage of the fact that f and g are separate. The *forward–backward* method [47, 48, 49, 50] is an example of a splitting method, since its iterations can be broken into a forward (gradient) step on f and a backward (proximal) step on g, performed consecutively:

Algorithm 3: Forward-backward algorithm with a fixed step-size parameter.

```
Choose $\mathbf{x}^0 \in \mathbb{R}^n,\ \tau > 0$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    $\mathbf{x}^{k+1} \leftarrow \operatorname{prox}_{\tau g}[\mathbf{x}^k - \tau\nabla f(\mathbf{x}^k)]$;
    $k \leftarrow k + 1$;
end
```

The splitting $f + g$ may not be unique. Different splittings lead to different versions of the forward–backward method. This method can be seen as a generalization of the two previous ones: when $g = 0$, we recover the gradient-descent method, while when $f = 0$, we recover the proximal-point method.

In order to prove under which conditions iterative algorithms such as the ones that we have been discussing solve the corresponding optimization problems, it can be useful to consider *fixed-point* methods. The set of fixed points of an operator $A : \mathcal{X} \to \mathcal{X}$ is

$$\operatorname{Fix} A \triangleq \{x \in \mathcal{X} \,|\, x = Ax\}. \tag{2.16}$$

If A is a Lipschitz-continuous operator, Fix A is closed [11, Proposition 4.14]. If A is nonexpansive, Fix A is closed and convex [11, Corollary 4.15]. Fixed-point methods try to find the fixed points of an operator (if they exist) by producing a sequence of points $\{x^k\}$ that should converge to one of them, given an initial point $x^0 \in \mathcal{X}$. Some iterative methods such as the gradient-descent and proximal-point ones are, in fact, fixed-point methods, in the sense that they are able to find fixed points of certain operators. When using these algorithms to solve optimization problems, the fixed points correspond to the solutions of the problem. We now briefly discuss how one can proceed in order to show that a method such as the forward–backward one converges to a solution of Problem (2.15).

It can be shown [51] that the forward–backward method produces a sequence of points that is *Fejér monotone* with respect to Fix $\operatorname{prox}_{\tau g}[\cdot - \tau\nabla f(\cdot)]$.[5] A sequence $\{x^k\}$ is said to be Fejér monotone with respect to a nonempty closed and convex set S in $\mathcal{X}$ if

$$\|x^{k+1} - \overline{x}\| \leq \|x^k - \overline{x}\|, \quad \forall \overline{x} \in S. \tag{2.17}$$

Such a sequence is bounded. Consequently, it possesses a subsequence that converges weakly to a point $x \in \mathcal{X}$. Such a point is said to be a *weak sequential cluster point* of $\{x^k\}$, and we denote the set of weak sequential cluster points of $\{x^k\}$ by W. Interestingly, it is also a consequence of Fejér monotonicity that a necessary and sufficient condition for the sequence $\{x^k\}$ to *converge weakly* to a point in S is that $W \subset S$ [51], [11, Chapters 2 and 5].[6]

Finally, it can also be shown (see Section 2.5) that x is a solution of Problem (2.15) if and only if

$$x = \operatorname{prox}_{\tau g}(x - \tau\nabla f(x)), \tag{2.18}$$

i.e., if and if $x \in \text{Fix}\ \{\operatorname{prox}_{\tau g}[\cdot - \tau\nabla f(\cdot)]\}$.

The interpretation of optimization algorithms as fixed-point methods and the conditions under which these methods can be used to find fixed points of an operator are discussed further in Section 2.5.

In Section 2.2, we discussed the use of second-order information to accelerate the progression towards a solution of the iterates produced by an algorithm. One may equally consider the use of other techniques. Relaxed steps [52, 53, 54] and inertial steps [55, 56, 57] are two possibilities. Both types of steps are characterized by the computation of a convex combination of two consecutive iterates. The weights of this combination can vary from iteration to iteration. Different weight combinations lead to different algorithms (with corresponding different convergence rates). We demonstrate the use of both techniques in the forward–backward method (cf. Algorithm 3).

The use of relaxed steps in the forward–backward method gives rise to the *relaxed forward–backward method*, which is

[5] This is a consequence of the fact that operator $\operatorname{prox}_{\tau g}[\cdot - \tau\nabla f(\cdot)]$ is averaged, a notion that is introduced in Section 2.5.

[6] It is sometimes useful to consider the notions of quasi-Fejér monotonicity and of Fejér monotonicity relative to a variable metric. A sequence is said to be quasi-Fejér monotone with respect to a nonempty closed and convex set S in $\mathcal{X}$ if

$$\|x^{k+1} - \overline{x}\| \leq \|x^k - \overline{x}\| + \epsilon^k, \quad \exists \left\{\epsilon^k\right\} \in \ell^1_+(\mathbb{N}), \quad \forall \overline{x} \in S.$$

A sequence is said to be Fejér monotone with respect to a nonempty closed and convex set S in $\mathcal{X}$ relative to a sequence $\left\{V^k\right\}$ such that

$$\begin{cases} V^k \in \mathcal{P}_\alpha(\mathcal{X}), \quad \alpha \in]0, +\infty[, \quad \forall k \in \mathbb{N}, \\ \sup_k \|V^k\| < \infty, \\ \left(1 + \eta^k\right) V^{k+1} \succeq V^k, \quad \left\{\eta^k\right\} \in \ell^1_+(\mathbb{N}), \quad \forall k \in \mathbb{N} \end{cases}$$

if

$$\|x^{k+1} - \overline{x}\|_{V^{k+1}} \leq (1 + \eta^k)\|x^k - \overline{x}\|_{V^k}, \quad \forall \overline{x} \in S.$$

Algorithm 4: Relaxed forward–backward method.

```
Choose $\mathbf{x}^0 \in \mathbb{R}^n,\ \tau > 0$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\lambda^k > 0$;
    $\mathbf{v}^k \leftarrow \mathbf{x}^k - \tau\nabla f(\mathbf{x}^k)$;
    $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \lambda^k(\mathrm{prox}_{\tau g}\mathbf{v}^k - \mathbf{x}^k)$;
    $k \leftarrow k + 1$;
end
```

The use of inertial steps in the forward–backward method gives rise to the following algorithm:

Algorithm 5: Forward-backward method with inertial steps.

```
Choose $\mathbf{y}^0 \in \mathbb{R}^n,\ \tau > 0$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\alpha^k > 0$;
    $\mathbf{x}^{k+1} \leftarrow \mathrm{prox}_{\tau g}[\mathbf{y}^k - \tau\nabla f(\mathbf{y}^k)]$;
    $\mathbf{y}^{k+1} \leftarrow \mathbf{x}^{k+1} + \alpha^k\left(\mathbf{x}^k - \mathbf{x}^{k+1}\right)$;
    $k \leftarrow k + 1$;
end
```

Both techniques can be combined [58, 59]. We recover the original forward–backward method by making $\lambda^k = 1$ for every k in Algorithm 4, or by making $\alpha^k = 0$ for every k in Algorithm 5. It is unclear which technique is more useful in practice, but Algorithm 5 has been proven to converge at an optimal rate in some sense [56].

It is also of interest to consider problems whose objective functions are given by the sum of two possibly nonsmooth functions. Consider the problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\mathrm{minimize}} \quad g(\mathbf{x}) + h(\mathbf{x}), \tag{2.19}$$

where g is defined as before and $h \in \Gamma_0(\mathbb{R}^n)$ is another possibly nonsmooth function. The possibility of g to be nonsmooth allows the use of nonsmooth data-fitting terms,[7] but prevents us from using the forward–backward method. A method that can be used is the relaxed Douglas–Rachford one, which is given by

[7] See Footnote 5 of Chapter 1.

Algorithm 6: Relaxed Douglas–Rachford method.

```
Choose $\mathbf{x}^0 \in \mathbb{R}^n,\ \tau > 0$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\lambda^k > 0$;
    $\mathbf{x}^k \leftarrow \operatorname{prox}_{\tau h} \mathbf{w}^k$;
    $\mathbf{v}^k \leftarrow \operatorname{prox}_{\tau g}(2\mathbf{x}^k - \mathbf{w}^k)$;
    $\mathbf{w}^{k+1} \leftarrow \mathbf{w}^k + 2\lambda^k(\mathbf{v}^k - \mathbf{x}^k)$;
    $k \leftarrow k + 1$;
end
```

The non-relaxed form of this method is obtained by making $\lambda^k = 1/2$ for every k.

Consider yet another problem,

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad g(\mathbf{x}) + h(\mathbf{D}\mathbf{x}), \tag{2.20}$$

where g and h are defined as before, and $\mathbf{D} \in \mathbb{R}^{m\times n}$. Problems such as these are discussed in Section 1.3 [cf. Problem (1.12)]. The matrix $\mathbf{D}$ can cause some difficulties when directly applying the Douglas–Rachford method to solve this problem, since that requires the computation of $\operatorname{prox}_{h(\mathbf{D}\cdot)}$, which may be intractable. For example, if h is the ℓ_1 norm, we lose the ability to compute its proximal operator element-wise.

One way to deal with this issue is to use some results from *duality theory*, in the sense of Fenchel, to formulate a problem that is equivalent to Problem (2.20), but that may be easier to solve. Given an optimization problem—usually termed *primal problem*—the Fenchel–Rockafellar duality theorem [11, Chapter 15] states that there is a *dual problem* whose minimum provides a lower bound on the minimum of the original problem (up to a minus sign). For problems of the form of Problem (2.20), if $g \in \Gamma_0(\mathbb{R}^n)$ and $h \in \Gamma_0(\mathbb{R}^m)$, then this bound is tight, under certain conditions.[8] In these cases, it is said that *strong duality* holds.

Closely related to these ideas is the concept of Legendre–Fenchel *conjugate* of a function f, which is defined as

$$f^* : \mathbb{R}^n \to [-\infty, +\infty] : \mathbf{x} \to \sup_{\mathbf{u}\in\mathbb{R}^n} \langle \mathbf{x}, \mathbf{u}\rangle - f(\mathbf{u}). \tag{2.21}$$

The conjugate is widely used in convex analysis and we recall some of its properties. Consider that $f \in \Gamma_0(\mathbb{R}^n)$. Then $f^* \in \Gamma_0(\mathbb{R}^n)$ and, by the Fenchel—Moreau theorem [11, Theorem 13.32], the biconjugate of f (the conjugate of the conjugate) is equal to f, i.e., $f^{**} = f$. Recalling the definition of the subdifferential [cf. Eq. (2.1], another property is that [11, Proposition 16.9]

$$\mathbf{u} \in \partial f(\mathbf{x}) \Leftrightarrow \mathbf{x} \in \partial f^*(\mathbf{u}), \quad \forall \mathbf{x}, \mathbf{u} \in \mathbb{R}^n. \tag{2.22}$$

[8] See Rockafellar [60, Corollary 31.2.1] and Bauschke and Combettes [11, Theorem 26.2] for some examples of such conditions. In finite-dimensional spaces, it is sufficient, e.g., to assume that there exists an $\mathbf{x} \in \text{ri }(\text{dom } g)$ such that $\mathbf{D}\mathbf{x} \in \text{ri }(\text{dom } h)$, where ri S is the relative interior [11, Definition 6.9] of a nonempty convex set $S \subset \mathbb{R}^n$. This condition is verified in many problems of interest in imaging.

or, in other words, $\partial f^* = (\partial f)^{-1}$. These definitions and properties can be used to show that the dual of Problem (2.20) is [11, Definition 15.19]

$$\underset{\mathbf{v}\in\mathbb{R}^m}{\text{minimize}} \quad g^*(-\mathbf{D}^*\mathbf{v}) + h^*(\mathbf{v}). \tag{2.23}$$

Recalling the definition of proximal operator [cf. Eq. (2.8)] as well, the notion of conjugate is also important in establishing the so-called *Moreau's decomposition*,

$$\operatorname{prox}_{\tau g}(\mathbf{x}) + \tau \operatorname{prox}_{g^*/\tau}(\mathbf{x}/\tau) = \mathbf{x}, \quad \forall\, \mathbf{x} \in \mathbb{R}^n. \tag{2.24}$$

Applying the Douglas–Rachford method to solve Problem (2.23) requires the computation of the proximal operator $\operatorname{prox}_{g^*(-\mathbf{D}^*\cdot)}$, which can be done using Eq. (2.24). This may be easier than the computation of $\operatorname{prox}_{h(\mathbf{D}\cdot)}$.

Some algorithms, instead of solving either the primal problem or the dual one, solve both of them. These algorithms compute a tentative solution to one of the problems and then use this solution to compute a tentative solution to the other. They are known as *primal–dual* methods, and we discuss two examples in what follows.

2.3.1. The alternating-direction method of multipliers and a primal–dual forward–backward-based method

Since $h \in \Gamma_0(\mathbb{R}^m)$, from the Fenchel-–Moreau theorem [11, Theorem 13.32], we have that $h = h^{**}$. By making use of Definition (2.21), Problem (2.20) can be rewritten as

$$\arg\min_{\mathbf{x}\in\mathbb{R}^n} \sup_{\mathbf{v}\in\mathbb{R}^m} \; g(\mathbf{x}) + \langle \mathbf{v}, \mathbf{D}\mathbf{x}\rangle - h^*(\mathbf{v}), \tag{2.25}$$

A solution pair of the latter problem, which we denote by $(\mathbf{x}, \mathbf{v})$, satisfies the relations [11, Theorem 26.2], [60, Theorem 31.3]

$$-\mathbf{D}^*\mathbf{v} \in \partial g(\mathbf{x}), \quad \mathbf{D}\mathbf{x} \in \partial h^*(\mathbf{v}). \tag{2.26}$$

Such a pair is usually known as a Karush-Kuhn-Tucker (KKT) pair.

In recent years, the alternating-direction method of multipliers (ADMM) and some other related algorithms have been widely used to solve several signal-processing and machine-learning problems due to the ability of these algorithms to transform complex optimization problems into a series of simpler subproblems.[9] ADMM can be viewed either as a primal–dual method (since it finds a KKT pair), as an application of the Douglas–Rachford method (cf. Algorithm 6) to solve Problem (2.23) [65], or as a Lagrangian method. In fact, another theory of duality that is intimately connected to Fenchel's duality is Lagrange's duality. Consider the equivalent formulation of Problem (2.20),

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} && g(\mathbf{x}) + h(\mathbf{u}) \\ &\text{subject to} && \mathbf{u} = \mathbf{D}\mathbf{x}, \end{aligned} \tag{2.27}$$

[9] See, e.g., [61, 62, 63, 64] for details, and see `http://stanford.edu/~boyd/admm.html` for a small list of applications of ADMM, along with more references.

where $\mathbf{u} \in \mathbb{R}^m$. Its *augmented Lagrangian* is

$$L_\gamma(\mathbf{x}, \mathbf{u}, \mathbf{d}) \triangleq g(\mathbf{x}) + h(\mathbf{u}) + \frac{\gamma}{2}\Big\|\mathbf{u} - \mathbf{D}\mathbf{x} - \mathbf{d}\Big\|^2, \tag{2.28}$$

where $\gamma > 0$ is a penalization parameter, and $\mathbf{d} \in \mathbb{R}^m$ is the *scaled* dual variable ($\mu\mathbf{d}$ is the dual variable). ADMM is as follows:

Algorithm 7: ADMM (scaled form).

```
Choose $\mathbf{u}^0 \in \mathbb{R}^m,\ \mathbf{d}^0 \in \mathbb{R}^m$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\gamma > 0$;
    $\mathbf{x}^{k+1} \leftarrow \underset{\mathbf{x}}{\arg\min} \quad L_\gamma(\mathbf{x}, \mathbf{u}^k, \mathbf{d}^k)$;
    $\mathbf{u}^{k+1} \leftarrow \underset{\mathbf{u}}{\arg\min} \quad L_\gamma(\mathbf{x}^{k+1}, \mathbf{u}, \mathbf{d}^k)$;
    $\mathbf{d}^{k+1} \leftarrow \mathbf{d}^k - (\mathbf{u}^{k+1} - \mathbf{D}\mathbf{x}^{k+1})$;
    $k \leftarrow k + 1$;
end
```

One may also solve problems of the form of Problem (2.3) when $f = 0$ and $g = 0$ by considering the use of ADMM to solve an alternative formulation of Problem (2.27) [64]. ADMM involves some operations with matrix $\mathbf{D}$ that might be hard to perform in large-scale problems if this matrix does not have an inherent exploitable structure. Alternatives to ADMM are, e.g., its linearized version (a particular case of a method known as proximal ADMM [66]) or methods such as the ones described next.

Consider now convex problems of the form of Problem (2.3) [Problem (1.15) is an instance of such a problem]. An algorithm that can be used to solve them is based on the forward–backward method, and is given by

Algorithm 8: Forward-backward-based primal–dual method.

```
Choose $\mathbf{x}^0 \in \mathbb{R}^n$, $\mathbf{d}_1^0 \in \mathbb{R}^{m_1},\ \ldots,\ \mathbf{d}_N^0 \in \mathbb{R}^{m_N},\ \tau > 0,\ \sigma_1 > 0,\ \ldots,\ \sigma_N > 0$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\lambda^k > 0$;
    $\mathbf{p}^k \leftarrow \mathrm{prox}_{\tau g}\left(\mathbf{x}^k - \tau\nabla f\left(\mathbf{x}^k\right) - \tau\sum_{j=1}^N \mathbf{D}_j^*\mathbf{d}_j^k\right)$;
    $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \lambda^k\left(\mathbf{p}^k - \mathbf{x}^k\right)$;
    for $j \leftarrow 1,\ \ldots,\ N$ do
        $\mathbf{q}_j^k \leftarrow \mathrm{prox}_{\sigma_j h_j^*}\left(\mathbf{d}_j^k + \sigma_j\mathbf{D}_j\left(2\mathbf{p}^k - \mathbf{x}^k\right)\right)$;
        $\mathbf{d}_j^{k+1} \leftarrow \mathbf{d}_j^k + \lambda^k\left(\mathbf{q}_j^k - \mathbf{d}_j^k\right)$;
    end
    $k \leftarrow k + 1$;
end
```

For details on the convergence conditions of this method, see Condat [67] and Vũ [68]. We recover the relaxed forward–backward method (cf. Algorithm 4) by making $N = 0$, and the relaxed Douglas–Rachford method (cf. Algorithm 6) by making $f = 0$, $N = 1$, $\mathbf{D}_1 = \mathbf{I}_n$, $\tau = 1/\sigma_1$, and $\lambda^k \to 2\lambda^k, \forall_k$.

The forward–backward method finds solutions to convex problems at a linear convergence rate under certain assumptions.[10] Similarly to the use of second-order information in the Newton method, we can also use second-order information in the forward–backward method. As an example, a Newton-type forward–backward method [69, 70] that minimizes the convex function $f + g$ is of the form of Algorithm 3 with Line 4 replaced by

$$\mathbf{x}^{k+1} \leftarrow \operatorname{prox}_g^{\mathbf{B}^k}\left(\mathbf{x}^k - \left[\mathbf{B}^k\right]^{-1} \nabla f\left(\mathbf{x}^k\right)\right), \tag{2.29}$$

where $\mathbf{B}^k \succ 0$ is the Hessian of f or an approximation of it. Such an algorithm only uses second-order information of function f (recall that f is differentiable everywhere, unlike g), but finds solutions at a superlinear or even quadratic local convergence rate (depending on the choice of $\mathbf{B}^k$) [70]. In Section 2.4, we discuss algorithms that use second-order information of the whole function, $f + g$. If we make $g = 0$, we recover a Newton-type method.

Variable-metric methods are generalizations of these algorithms. In these methods, the matrix $\mathbf{B}^k$ need not to be the Hessian nor an approximation of it. For example, $\mathbf{B}^k$ can be a preconditioning matrix [71, 72, 53]. A variable-metric version of Algorithm 8 is

Algorithm 9: Forward-backward-based variable-metric primal–dual method.

```
Choose $\mathbf{x}^0 \in \mathbb{R}^n,\ \mathbf{d}_1^0 \in \mathbb{R}^{m_1},\ \ldots,\ \mathbf{d}_N^0 \in \mathbb{R}^{m_N}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\lambda^k > 0,\ \mathbf{U}^k \succ 0$;
    $\mathbf{p}^k \leftarrow \operatorname{prox}_g^{(\mathbf{U}^k)^{-1}}\left(\mathbf{x}^k - \mathbf{U}^k\left(\nabla f\left(\mathbf{x}^k\right) + \sum_{j=1}^N \mathbf{D}^*\mathbf{d}_j^k\right)\right)$;
    $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \lambda^k\left(\mathbf{p}^k - \mathbf{x}^k\right)$;
    for $j \leftarrow 1,\ \ldots,\ N$ do
        Choose $\mathbf{U}_j^k \succ 0$;
        $\mathbf{q}_j^k \leftarrow \operatorname{prox}_{h_j^*}^{(\mathbf{U}_j^k)^{-1}}\left(\mathbf{d}_j^k + \mathbf{U}_j^k\mathbf{D}_j\left(2\mathbf{p}^k - \mathbf{x}^k\right)\right)$;
        $\mathbf{d}_j^{k+1} \leftarrow \mathbf{d}_j^k + \lambda^k\left(\mathbf{q}_j^k - \mathbf{d}_j^k\right)$;
    end
    $k \leftarrow k + 1$;
end
```

[10] Namely, that the objective function is strongly convex (see Footnote 2).

2.4. Semismooth Newton methods

We now discuss one last class of algorithms, known as the class of *semismooth Newton* methods. These methods were originally developed with the goal of using a Newton-like method to minimize nonsmooth functions. In this section, we briefly discuss the use of semismooth Newton methods to solve Problem (1.11), which, we recall, is

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2+\mu\|\mathbf{x}\|_1. \tag{2.30}$$

For problems such as these, it was shown by Hintermüller [73] that certain semismooth Newton methods are equivalent to some *active-set* methods (see below for a definition). Although active-set methods are normally used to solve linear and quadratic problems, they can also be used to solve problems involving sparsity-inducing regularizers, such as the ℓ_1 norm. In fact, active-set methods are particularly suited to these problems, since they are able to take computational advantage of the sparsity of the variable being minimized. The discussion that follows also holds for data-fitting terms other than $\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2$, as long as they are differentiable.

We start by discussing active-set methods. The minimizer of Problem (2.30) is assumed to be sparse. If we know which of its entries are zero, instead of solving Problem (2.30), we can solve an equivalent problem. Let $\hat{\mathbf{x}}\in\mathbb{R}^n$ be a solution to Problem (2.30), let $\hat{A}$ denote the set comprising the indices of the entries of $\hat{\mathbf{x}}$ that are zero, and let $\hat{I}$ denote the set comprising the remaining indices. It is clear that the problem

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} && \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2+\mu\|\mathbf{x}\|_1 \\ &\text{subject to} && [\mathbf{x}]_i=0,\, i\in\hat{A} \end{aligned} \tag{2.31}$$

has the same set of solutions as Problem (2.30). However, this new problem has a much smaller number of variables to minimize, which usually means that it is much faster to solve.

In practice, we do not know beforehand which entries of $\hat{\mathbf{x}}$ are zero. Active-set methods address this issue by finding estimates of the sets $\hat{A}$ and $\hat{I}$ by following some predefined strategy (see below). We denote these estimates by A^1 and I^1, respectively. These methods then produce a solution to Problem (2.31) with $\hat{A}$ replaced by A^1. We denote this solution by $\mathbf{x}^1$. Since the set A^1 is not guaranteed to be the same as $\hat{A}$, a new estimate A^2 is produced in light of the solution $\mathbf{x}^1$, and a new solution $\mathbf{x}^2$ is produced based on A^2. These operations are then repeated until a certain stopping criterion is satisfied. The successive estimates of the set $\hat{A}$ are called the *active sets* and are denoted by A^k. The variables which are not indexed by the active sets are called *free* variables and are indexed by the set I^k. The successive solutions to Problem (2.31) can be found by any optimization algorithm, and, in this sense, active-set methods are meta-methods. A common strategy is to solve this problem via a proximal-Newton method [74, Section 8.2].

The choice of strategy determining how both sets are estimated yields different algorithms. In what follows, we discuss an example of such a strategy by showing how some semismooth Newton methods can be seen as active-set methods.

As discussed in Section 2.2, the Newton method can be used to minimize a smooth function $f \in \Gamma_0(\mathbb{R}^n)$ by iterating $\mathbf{x}^{k+1} = \mathbf{x}^k - \lambda^k[\nabla^2 f(\mathbf{x}^k)]^{-1}\nabla f(\mathbf{x}^k)$. It can also be used to solve smooth nonlinear equations, in the sense that these equations can be written as the zero of smooth functions. Let $\mathbf{u} \in \mathbb{R}^n$, and let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a continuously differentiable operator. The solution to the equation $F(\mathbf{u}) = \mathbf{0}$ via the Newton method is found by iterating $\mathbf{u}^{k+1} = \mathbf{u}^k - [F'(\mathbf{u}^k)]^{-1}F(\mathbf{u}^k)$, where F' denotes the Fréchet derivative of F [11, Definition 2.45]. It is clear that if $n = 1$ and if F is the gradient of a function f, solving $F(\mathbf{u}) = \mathbf{0}$ is equivalent to minimizing f.

The minimization problems discussed in the previous sections can be written as nonlinear equations. Take Problem (2.30), for example. Its solution should satisfy the fixed-point equation [cf. Eq. (2.18)]

$$\mathbf{x} = \operatorname{prox}_{\mu\tau\|\mathbf{x}\|_1}\left(\mathbf{x} - 2\tau\mathbf{H}^*(\mathbf{H}\mathbf{x} - \mathbf{y})\right). \tag{2.32}$$

By defining $G : \mathbb{R}^n \to \mathbb{R}^n : \mathbf{u} \to \mathbf{u} - \operatorname{prox}_{\mu\tau\|\mathbf{u}\|_1}(\mathbf{u} - 2\tau\mathbf{H}^*(\mathbf{H}\mathbf{u} - \mathbf{y}))$, Eq. (2.32) can be written as the nonlinear equation

$$G(\mathbf{x}) = \mathbf{0}. \tag{2.33}$$

This equation is not smooth, since the soft-thresholding operator [cf. Eq. (2.12)] is not differentiable everywhere. For this reason, the ordinary Newton method cannot be used to find a solution to Eq. (2.33). However, there are nonsmooth versions of this method. In fact, semismooth Newton methods can be used to minimize semismooth functions, which were first defined in finite-dimensional spaces by Mifflin [75]. Examples of semismooth functions are the Euclidean norm and piecewise-differentiable functions, such as Eq. (2.12) [76, Chapter 2]. Semismooth functions are not smooth, and, in order to be able to use the Newton method to find their minimizers, Mifflin used the notion of generalized gradient introduced by Clarke [77], which, unlike the subdifferential of a convex function, is defined for functions that are locally Lipschitz continuous and that may not be convex. Semismooth Newton methods are able to find minimizers at a superlinear convergence rate (locally and under certain conditions).

These methods were extended by Chen et al. [78] to the problem of finding solutions of operator equations in Banach spaces. Chen et al. introduced the notion of *slant differentiability* of operators. It can be defined as follows [78, Definitions 2.1 and 2.2]: Let $\mathcal{X}$ and $\mathcal{V}$ be Banach spaces, and let $G : D \subset \mathcal{X} \to \mathcal{V}$ be an operator. G is said to be slantly differentiable in the open domain $U \subset D$ if there exists an operator $H : D \to \mathcal{L}(\mathcal{X}, \mathcal{V})$ such that the family of operators $\{H(x+d)\}$ is uniformly bounded in the operator norm[11] for sufficiently small d and

$$\lim_{d \to 0} \frac{G(x+d) - G(x) - H(x+d)d}{\|d\|} = 0 \tag{2.34}$$

for every $x \in U$. The operator H is said to be a *slanting function* for G in U and may not be unique. An operator G is slantly differentiable at x if and only if G is Lipschitz continuous at x [78, Theorem 2.6].

[11] See, e.g., [11, Lemma 2.16] for a definition of uniform boundedness.

Suppose now that we want to solve the equation $G(x) = 0$ and that G is slantly differentiable at a solution $\hat{x} \in \mathcal{X}$ of this equation. Let $C > 0$, let H be a slanting function for G at $\hat{x}$, and let $\|[H(x)]^{-1}\| \leq C$ in a neighborhood of $\hat{x}$. Given a linear operator $V \in \mathcal{L}(\mathcal{X}, \mathcal{V})$, the algorithm

Algorithm 10: Semismooth Newton method.

```
1 Choose $x^0 \in \mathcal{X}$ and $V$ such that $\|V(\hat{x}+d) - H(\hat{x}+d)\| \to 0$ as $\|d\| \to 0$;
2 $k \leftarrow 1$;
3 while stopping criterion is not satisfied do
4     $x^{k+1} \leftarrow x^k - [V(x^k)]^{-1}\, G(x^k)$;
5     $k \leftarrow k+1$;
6 end
```

locally converges at a superlinear rate to $\hat{x}$ [78, Theorem 3.4].

It can be shown that the operator G of Eq. (2.33) is slantly differentiable [79]. Consequently, we can use Algorithm 10 to find a solution to Eq. (2.33). In fact, Line 4 of that algorithm can be rewritten[12] for every k as

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - [V(\mathbf{x}^k)]^{-1} \left(\mathbf{x}^k - \operatorname{prox}_{\mu\tau\|\cdot\|_1}\left(\mathbf{x}^k - 2\tau\mathbf{H}^*(\mathbf{H}\mathbf{x}^k - \mathbf{y})\right)\right). \tag{2.35}$$

A possible choice [79] for $V(x^k)$ that satisfies the conditions enunciated in Line 1 of Algorithm 10 is given, in matrix notation, by

$$\begin{bmatrix} \tau[\mathbf{H}]^*_{:I^k}[\mathbf{H}]_{:I^k} & \tau[\mathbf{H}]^*_{:I^k}[\mathbf{H}]_{:A^k} \\ \mathbf{0} & \mathbf{I}_{\operatorname{card}(A^k)}. \end{bmatrix}, \tag{2.36}$$

where $\operatorname{card}(\cdot)$ denotes the cardinality of a set and

$$\begin{aligned} A^k &\triangleq \{i \in \mathbb{N} : |[\mathbf{x}^k - 2\tau\mathbf{H}^*(\mathbf{H}\mathbf{x}^k - \mathbf{y})]_i| \leq \tau\mu\}, \\ I^k &\triangleq \{i \in \mathbb{N} : |[\mathbf{x}^k - 2\tau\mathbf{H}^*(\mathbf{H}\mathbf{x}^k - \mathbf{y})]_i| > \tau\mu\}. \end{aligned} \tag{2.37}$$

Griesse and Lorenz [79] showed that this algorithm is equivalent to an active-set method by rewriting the iteration (2.35) as

$$\mathbf{x}^{k+1} \leftarrow \begin{bmatrix} \left([\mathbf{H}]^*_{:I^k}[\mathbf{H}]_{:I^k}\right)^{-1} [\mathbf{H}^*\mathbf{y} + \mathbf{e}^k_{\pm}\mu]_{I^k} \\ \mathbf{0} \end{bmatrix}, \tag{2.38}$$

where, for every k,

$$\mathbf{e}^k_{\pm} \triangleq \operatorname{sgn}\left[\mathbf{x}^k - 2\tau\mathbf{H}^*(\mathbf{H}\mathbf{x}^k - \mathbf{y})\right]. \tag{2.39}$$

The dimension of the problem to solve at each iteration k is given by $\operatorname{card}(A^k)$. Naturally, the sparser the solution is estimated to be, the smaller the dimension of this problem is. This is the reason why these methods are able to achieve faster convergence rates in practice than others such as ADMM, since they require the solution of a problem involving the full matrix $\mathbf{H}^*\mathbf{H}$.

[12] Note that Problem (2.30) is set in a finite-dimensional space.

2.5. Operator theory

Functional analysis and, in particular, the theory of monotone and of nonexpansive operators (see below for the definitions) can be used to derive and prove a number of interesting properties of many of the algorithms used to find the solutions of convex optimization problems [80]. An operator $A : \mathcal{X} \to 2^{\mathcal{X}}$ is said to be *monotone* if

$$\langle u - v, x - y \rangle \geq 0, \quad \forall\, (x, u) \in \operatorname{gra} A, \quad \forall\, (y, v) \in \operatorname{gra} A. \tag{2.40}$$

An operator is *maximally monotone* if there exists no other monotone operator whose graph properly contains gra A. As an example, if $g \in \Gamma_0(\mathcal{X})$, than ∂g is maximally monotone [11, Theorem 20.40]. Let $\beta \in]0, +\infty[$. We say that an operator is *strongly monotone* with constant β if $A - \beta \mathrm{Id}$ is monotone.

Monotone operators are connected to optimization problems as follows. Take, for example, Problem (2.15). According to Fermat's rule, its solutions should satisfy the inclusion $0 \in \nabla f(x) + \partial g(x)$. Consequently, solving Problem (2.15) can be seen as a particular case of the problem of finding a zero of the sum of two monotone operators A and C acting on a Hilbert space $\mathcal{X}$, i.e.,

$$\text{find } x \in \mathcal{X} \quad \text{such that } 0 \in Ax + Cx, \tag{2.41}$$

if ones makes $A = \nabla f$ and $C = \partial g$.

We now list some properties of operators. Let D be a nonempty set of $\mathcal{X}$ and let $R : D \to \mathcal{X}$ be a nonexpansive operator. We say that an operator $A : D \to \mathcal{X}$ is λ-*averaged* if there exists $\lambda \in]0, 1[$ such that

$$A = (1 - \lambda)\mathrm{Id} + \lambda R. \tag{2.42}$$

An averaged operator A obeys the following contractive property [11, Proposition 4.25]:

$$\begin{aligned} \|Ax - Ay\|^2 &\leq \|x - y\|^2 \\ &\quad - \frac{1-\lambda}{\lambda} \|(\mathrm{Id} - A)\, x - (\mathrm{Id} - A)\, y\|^2, \quad \forall x \in D, \quad \forall y \in D. \end{aligned} \tag{2.43}$$

When $\lambda = 1/2$, A is said to be *firmly nonexpansive.* Proximal operators [cf. Eq. (2.8)] are examples of firmly nonexpansive operators [11, Corollary 23.8].

Let $\beta \in]0, +\infty[$. We say that a (single-valued) operator $C : D \to \mathcal{X}$ is β-*cocoercive* if

$$\langle Cx - Cy, x - y \rangle \geq \beta \|Cx - Cy\|^2, \quad \forall\, x \in D, \forall\, y \in D. \tag{2.44}$$

An operator C is β-cocoercive if and only if βA is $\frac{1}{2}$-averaged [11, Remark 4.24(iv)]. Let $f \in \Gamma_0(\mathcal{X})$ and let ∇f be β-Lipschitz continuous. Then, according to the Baillon—Haddad theorem [11, Corollary 18.16], ∇f is $\frac{1}{\beta}$-cocoercive.

The zeros of a monotone operator can be found by using fixed-point methods on appropriate operators. Let $A : \mathcal{X} \to 2^{\mathcal{X}}$ be a maximally monotone operator and assume that zer $A \neq \emptyset$. Associated to this operator is its *resolvent*

$$J_{\tau A} \triangleq (\mathrm{Id} + \tau A)^{-1}. \tag{2.45}$$

The set of fixed points of $J_{\tau A}$ coincides with the set of zeros of A [11, Proposition 23.38]. In order to find one of these zeros, one may consider an iteration of the form

$$x^{k+1} = J_{\tau A} x^k, \quad x^0 \in \mathcal{X}. \tag{2.46}$$

Algorithm 2 is a particular case of this iteration if one makes $A = \partial g$, since $J_{\tau \partial g} = \operatorname{prox}_{\tau g}$ [11, Example 23.3].

In general, if one wants to find a fixed point of a nonexpansive operator R, iterations of the form $x^{k+1} = Rx^k$—which are known as *Banach–Picard iterations*—may fail to find such a point. An alternative is the so-called *Krasnosel'skiĭ–Mann* method:

Algorithm 11: Krasnosel'skiĭ–Mann method.

```
Choose $x^0 \in \mathcal{X}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\lambda^k > 0$;
    $x^{k+1} \leftarrow x^k + \lambda^k (Rx^k - x^k)$;
    $k \leftarrow k + 1$;
end
```

Under certain conditions, this algorithm converges weakly to Fix R even when R is merely nonexpansive. The operator $T^k \triangleq \mathrm{Id} + \lambda^k(R - \mathrm{Id})$ is a λ^k-averaged operator and, consequently, this method can be seen as an iteration of averaged operators.

We now present two algorithms that can be used to solve two different inclusion problems. The first algorithm can be used to solve the inclusion $0 \in Ax + Cx$, where A is a maximally monotone operator and C is a β-cocoercive operator. It is given by

Algorithm 12: Relaxed forward–backward method (with operators).

```
Choose $x^0 \in \mathcal{X}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\lambda^k > 0$;
    $v^{k+1} \leftarrow x^k - \tau C x^k$;
    $x^{k+1} \leftarrow x^k + \lambda^k (J_{\tau A} v^{k+1} - x^k)$;
    $k \leftarrow k + 1$;
end
```

Algorithm 4 is a particular case of this algorithm, with $A = \partial g$ and $C = \nabla f$. Algorithm 12, in turn, is a particular case of Algorithm 11, with $R = J_{\tau A} \circ (\mathrm{Id} - \tau C)$.

The second algorithm can be used to solve the inclusion $0 \in Ax + Bx$, where A and B are maximally monotone operators. It is given by

Algorithm 13: Relaxed Douglas–Rachford method (with operators).

```
Choose $x^0 \in \mathcal{X}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\lambda^k > 0$;
    $v^k \leftarrow J_{\tau B} x^k$;
    $w^k \leftarrow J_{\tau A}(2v^k - x^k)$;
    $x^{k+1} \leftarrow x^k + 2\lambda^k(w^k - v^k)$;
    $k \leftarrow k + 1$;
end
```

Algorithm 6 is a particular case of this algorithm, with $A = \partial g$ and $B = \partial h$. Algorithm 13, in turn, is a particular case of Algorithm 11, with $R = (2J_{\tau A} - \mathrm{Id}) \circ (2J_{\tau B} - \mathrm{Id})$.

Many other algorithms can also be shown to be particular cases of Algorithm 11. In fact, all the algorithms discussed in this chapter are particular cases of this algorithm, with the exception of the ones discussed in Section 2.4. There are at least two advantages in looking at convex problems and at these algorithms as particular cases of monotone inclusions and of fixed-point methods, respectively: (a) the analysis of the convergence of these algorithms becomes easier, since one is able to take advantage of the many results on operator theory that are available from the literature, and (b) besides convex problems, other problems of interest can also be seen as particular cases of monotone inclusions.[13]

In Section 2.3, we described how a preconditioning matrix can be used to increase the convergence rate of algorithms such as the proximal-point method. In what follows, we show how a preconditioning operator can be used with a similar goal in the forward–backward method.

Recall that the forward–backward method can be used to solve the problem

$$\text{find } x \in \mathcal{X} \quad \text{such that } 0 \in Ax + Cx, \tag{2.47}$$

which has the same set of solutions as the problem

$$\text{find } x \in \mathcal{X} \quad \text{such that } 0 \in UAx + UCx, \tag{2.48}$$

where $U \succ 0$. It may be convenient to solve Problem (2.48) instead of Problem (2.47), and one can also use the forward–backward method to solve Problem (2.48). In this case, and in view of Algorithm 11, the nonexpansive operator is $R = J_{\tau UA} \circ (\mathrm{Id} - \tau UC)$. Furthermore, the operator U can be made to vary in each iteration, yielding a variable-metric method [81].

[13] Variational-inequalities problems are an example of such problems. See, e.g., [11, Definition 25.12, Proposition 25.18 and Example 25.19].

2.6. Comments and references

All the algorithms discussed in this chapter, with the exception of semismooth Newton ones, can be seen as instances of the Krasnosel'skiĭ–Mann method (Algorithm 11). In Chapter 5, we propose and discuss some properties of an extension of this method, and show that semismooth Newton methods are instances of it.

Classical references on the topics of convex and of numerical optimization are Rockafellar [60], Nesterov [40], Nocedal and Wright [39], and the more recent Bauschke and Combettes [11]. For more details on optimization algorithms applied to signal and imaging problems, see Burger et al. [82], Chambolle and Pock [83], Combettes et al. [84], and Combettes and Pesquet [44]. For more details on optimization methods in a general context, see Bottou et al. [74], Komodakis and Pesquet [43], Parikh and Boyd [85], and Ryu and Boyd [52]. For more details on semismooth Newton methods, see Hintermüller [86] and Ulbrich [12, 76].

Part II.

Contributions

3. A framework for fast image deconvolution with incomplete observations

This chapter describes the contributions of the writer in Simões et al. [9]. We start by making some practical remarks about the problem of image deconvolution, and by briefly describing the related problems of image superresolution and of image inpainting with deblurring. We then proceed to present the contributions of this work.

Contents

3.1. Introduction

Consider the estimation of a sharp image from a blurred one. Let $b \in \mathbb{N}$ and assume that the support of the convolution filter has size $(2b+1) \times (2b+1)$ pixels and is centered at the origin. Let the size of the blurred image be $m \times n$, with $m, n \in \mathbb{N}$. To express that image as a function of the sharp one, we need to consider a region of the sharp image of size $m' \times n'$, with $m' = m + 2b$ and $n' = n + 2b$; the central $m \times n$ zone of this sharp image is in the same spatial location as the blurred image. Fig. 3.1 illustrates this situation.

As discussed in Section 1.2, we can express the blurring operation via a linear model of the form

$$\mathbf{y} = \mathbf{T}\mathbf{x} + \mathbf{n}, \tag{3.1}$$

where $\mathbf{y} \in \mathbb{R}^{mn}$ is the observed image, $\mathbf{x} \in \mathbb{R}^{m'n'}$ is the sharp image, $\mathbf{T} \in \mathbb{R}^{mn \times m'n'}$ is a block-Toeplitz-Toeplitz-block (BTTB) matrix such that $\mathbf{Tx}$ represents the convolution of the sharp image with the blurring point-spread function (PSF), and $\mathbf{n} \in \mathbb{R}^{mn}$ is the observation noise.

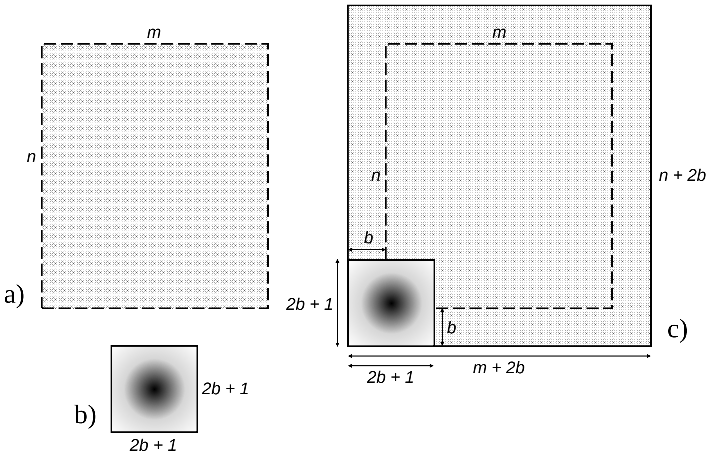


Figure 3.1.: An illustration of the dimensions of the images involved in typical deblurring problems. a) Blurred image. b) Blurring filter. c) Sharp image.

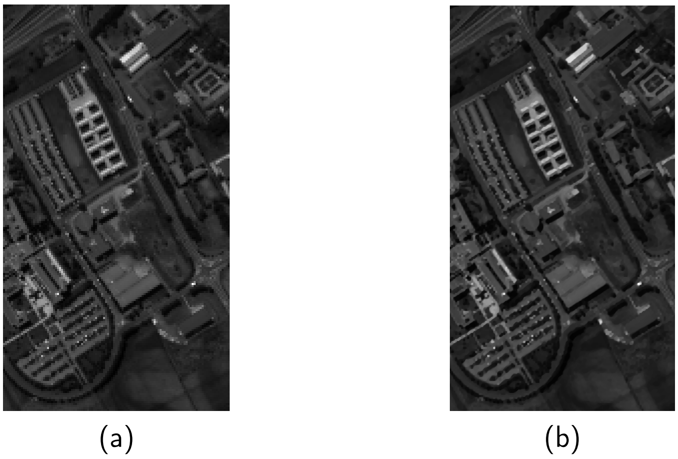


Figure 3.2.: Image superresolution: original (a) and superresolved (b) images.

In image deblurring problems, some of the pixels of the blurred image may not be observed. Examples of such pixels are the pixels corresponding to (a) saturated or missing pixels, (b) the extra pixels that would have been observed if the sensors that acquired the image had a higher spatial resolution, and (c) the boundaries of the image. The estimation of the pixels referred to in points (a) and (b) is related to the problems of image inpainting with deblurring and of image superresolution, respectively. Image superresolution is characterized by the generation of high-resolution images from low-resolution ones—see Fig. 3.2 for an example. This problem can be addressed within the framework of inverse problems. Let $\mathbf{y}_l$ denote a given low-resolution image. A possible way to relate it to its high-resolution version is to consider that $\mathbf{y}_l$ is a degraded version of an image $\mathbf{x} \in \mathbb{R}^{mn}$, to be estimated. With this in mind, superresolution can be modeled

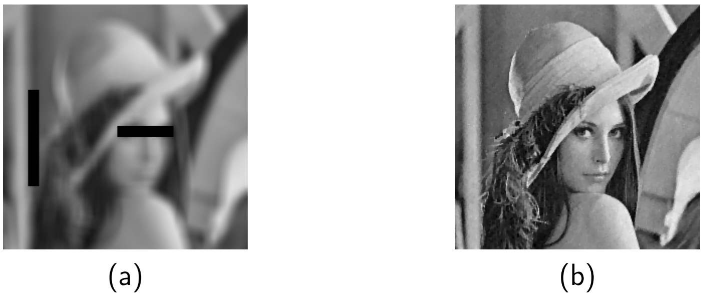


Figure 3.3.: Image inpainting with deblurring: observed (a) and estimated (b) images. In the observed image, the black rectangles correspond to unobserved regions.

as

$$\mathbf{y}_l = \mathbf{STx} + \mathbf{n}, \tag{3.2}$$

where $\mathbf{y}_l \in \mathbb{R}^{\frac{mn}{r^2}}$ is the observed image, $r \in \mathbb{N}$ denotes a spatial downsampling factor, and $\mathbf{S} \in \mathbb{R}^{\frac{mn}{r^2} \times mn}$ is a sampling matrix whose columns are a subset of the columns of the identity matrix. In this context, $\mathbf{T}$ is normally chosen to correspond to an anti-aliasing filter. We can obtain an estimate of $\mathbf{x}$ by solving a problem of the form of (1.9) with the data-fitting term based on Eq. (3.2) instead of on Eq. (1.1). A related problem is image inpainting, which can be coupled with image deblurring. Imagine that, for some reason, there are missing pixels in a given image. Image inpainting, per se, refers to the problem of recovering those pixels. If the observed image is also blurred, one faces the problem of image inpainting with deblurring: as an example, consider Fig. 3.3. In this case, model (3.2) can still be used, but $\mathbf{S}$ is the identity matrix with some of its rows eliminated, namely the ones corresponding to the elements of $\mathbf{x}$ that are missing in $\mathbf{y}_l$, and $\mathbf{T}$ corresponds to the blurring filter . Regarding point (c), above, consider again Fig. 3.1. Often, in simple deconvolution problems, one is only interested in estimating the central $m \times n$ region of $\mathbf{x}$, which we shall designate by *cropped sharp image*, denoted by $\bar{\mathbf{x}}$. If one is interested in estimating the whole image $\mathbf{x}$, this can be seen as an inpainting problem, since we are estimating the pixels of $\mathbf{x}$ in a boundary zone of width b around the central $m \times n$ region, and this zone is not present in $\mathbf{y}$. Even if we do not wish to estimate this zone, the need to properly handle the boundary zone still exists in most real-life deconvolution methods, as we discuss below.

In imaging problems, one is faced with the prospect of dealing with millions of variables. As an example, an optimization problem of the form of (1.9) involving a 1024×1024 image has 1,048,576 variables. To explicitly store in a computer the convolution matrix $\mathbf{T}$ that is associated with such an image would require vast amounts of memory. Fortunately, as mentioned in Section 1.2, these matrices are highly structured, and, under certain assumptions, it is possible to take advantage of this fact to store them with the same memory requirements as those of an image. Nevertheless, a difficulty with the use of model (3.1) is that deconvolution methods based on it normally involve products by large BTTB matrices and/or the inversion of such matrices, and both operations are computationally heavy. In order to obtain fast deconvolution methods, many authors replace model (3.1) with models that involve simpler computations. One of the most frequently used models is

$$\mathbf{y} = \overline{\mathbf{T}}\bar{\mathbf{x}}' + \mathbf{n}, \tag{3.3}$$

in which $\overline{\mathbf{T}}$ is a block-circulant-circulant-block (BCCB) matrix of size $mn \times mn$, and $\bar{\mathbf{x}}' \in \mathbb{R}^{mn}$ is an approximation of the true cropped sharp image $\bar{\mathbf{x}}$. $\overline{\mathbf{T}}\bar{\mathbf{x}}'$ represents the circular convolution of $\bar{\mathbf{x}}'$ with the blurring PSF. The fact that $\bar{\mathbf{x}}'$ is only an approximation of the true image $\bar{\mathbf{x}}$ means that model (3.3) is not an exact model of the convolution process. As a consequence of this, the sharp images obtained by these methods normally exhibit artifacts, typically in the form of ringing. The use of model (3.3) is often referred to as the use of *periodic boundary conditions*, because it is equivalent to the use of model (3.1) with the true sharp image $\mathbf{x}$ replaced with an image obtained by periodically repeating $\bar{\mathbf{x}}$ in the horizontal and vertical directions, and then retaining only the central $m' \times n'$ region of the resulting periodic image. Other possibilities exist for obtaining fast deconvolution methods, besides the use of periodic boundary conditions. For example, one can use *reflexive* or *anti-reflexive* boundary conditions, which, under appropriate assumptions, lead to matrices that are diagonalizable, respectively, by the 2-D discrete cosine transform and the 2-D discrete sine transform, and therefore also yield significant speed advantages [87, 14, 88]. However, the use of any of these boundary conditions (and, in fact, the use of any artificially imposed boundary conditions) corresponds to the use of an inexact convolution model, and therefore gives rise to artifacts.

The speed advantage of using model (3.3) comes from the fact that BCCB matrices have a particular spectral decomposition, $\mathbf{T} = \mathbf{F}^*\mathbf{\Lambda}\mathbf{F}$, where $\mathbf{F}$ is the 2-D discrete Fourier transform (DFT) unitary matrix, and the elements of the diagonal matrix $\mathbf{\Lambda}$ are the coefficients of the DFT of the blurring PSF. Matrix-vector multiplications with $\mathbf{F}$ and $\mathbf{F}^*$ can be performed with the fast Fourier Transform (FFT), which is an algorithm that computes the DFT of a vector in an efficient way. It uses a "divide-and-conquer" strategy that takes advantage of the inherent redundancy of the Fourier transform. For a generic $\mathbf{x}$, the use of the FFT allows one to reduce the number of operations required to compute $\mathbf{T}\mathbf{x}$ from $O[(mn)^2]$ to $O[mn\log(mn)]$, and the number of operations required to compute $\mathbf{T}^{-1}\mathbf{x}$ from $O[(mn)^3]$—as is the case when one inverts a generic matrix—to $O[mn\log(mn)]$. As an example of the use of the FFT, consider a version of Problem (1.6) that is under the assumption that $\mathbf{T}$ is BCCB, i.e., a version with the data-fitting term based on model (3.3). The solution to this problem is given by

$$\bar{\mathbf{x}}' = \mathcal{F}^{-1}\left\{(\mathcal{F}^*\{\mathbf{t}\} \otimes \mathcal{F}\{\mathbf{y}\}) \oslash (\mathcal{F}^*\{\mathbf{t}\} \otimes \mathcal{F}\{\mathbf{t}\} + \mu)\right\}, \tag{3.4}$$

where $\otimes$ and $\oslash$ denote element-wise multiplication and division, respectively, $\mathcal{F}\{\cdot\}$ and $\mathcal{F}^{-1}\{\cdot\}$ denote the direct and inverse DFT, respectively, $\cdot^*$ denotes the conjugate operator, and $\mathbf{t} \in \mathbb{R}^{mn}$ is a vector representing the lexicographical ordering of the filter.[1]

The occurrence of artifacts can be completely eliminated by the use of an exact model of the convolution process. A relatively recent method, which we shall designate by

[1] Matrix $\mathbf{T}$ and vector $\mathbf{t}$ both refer to the same object (the columns of $\mathbf{T}$ are permutations of $\mathbf{t}$).

AM [89, 90], uses a model of the form

$$\mathbf{y} = \mathbf{M}\widetilde{\mathbf{T}}\mathbf{x} + \mathbf{n}, \tag{3.5}$$

in which $\widetilde{\mathbf{T}}$ is an $m'n' \times m'n'$ BCCB matrix that corresponds to a circular convolution with the blurring PSF, and $\mathbf{M}$ is an $mn \times m'n'$ masking matrix that selects, from $\widetilde{\mathbf{T}}\mathbf{x}$, only the pixels that correspond to the observed image, discarding a boundary zone of width b in the periphery of the image $\widetilde{\mathbf{T}}\mathbf{x}$. The circular convolution $\widetilde{\mathbf{T}}\mathbf{x}$ in Eq. (3.5) only differs from the linear (i.e., non-circular) convolution $\mathbf{T}\mathbf{x}$ of Eq. (3.1) by the presence of that boundary zone, and therefore Eq. (3.5) is an exact model of the convolution process. Computationally, this method has the advantage of using a diagonalizable matrix, $\widetilde{\mathbf{T}}$, but needs to deal with the fact that $\mathbf{M}\widetilde{\mathbf{T}}$ is not easily diagonalizable. This difficulty is circumvented, in AM, through an adaptation of alternating-direction method of multipliers (ADMM) [64, 91]. By means of the splitting of a variable, AM decouples the matrix $\mathbf{M}$, which is diagonalizable in the spatial domain, from $\widetilde{\mathbf{T}}$, which is diagonalizable in the frequency domain, thereby allowing a significant speedup to be achieved. The matrix $\mathbf{M}$ can also be used to accommodate incomplete observations on $\mathbf{x}$, similarly to what is done in Eq. (3.2) with matrix $\mathbf{S}$.

3.1.1. Contributions and outline

In Simões et al. [9], we propose a new framework for solving deconvolution problems with unobserved pixels. This framework is an efficient, high-quality alternative to the use of heuristic methods—such as edge tapering [92]—to reduce the artifacts produced by deconvolution methods that assume periodic boundary conditions. It can be used with any fast deconvolution method, irrespective of the specific boundary conditions that it assumes. An example of how this framework can be used to extend an off-the shelf, state-of-the art deconvolution method to the use of unknown boundaries is also discussed. Furthermore, we propose a specific implementation of this framework, based on ADMM. We provide a proof of convergence for the resulting algorithm, which can be seen as a “partial” ADMM, in which not all variables are dualized. We report experimental comparisons with other primal–dual methods, where the proposed one performs at the level of the state of the art. Four different kinds of applications are tested in the experiments: deconvolution, deconvolution with inpainting, superresolution, and demosaicing, all with unknown boundaries.

This chapter describes the contributions of that work and presents some of the experimental results. It is structured as follows. Section 3.2 describes the framework used to solve deconvolution problems with unobserved pixels. Section 3.3 discusses the “partial” ADMM method. Section 3.4 details where that work was made available to the research community. Section 3.5 concludes.

3.2. Proposed framework

We propose a deconvolution framework that uses a different convolution model from (3.5). The convolution process is modeled as

$$\tilde{\mathbf{y}} = \widetilde{\mathbf{T}}\mathbf{x} + \mathbf{n}, \tag{3.6}$$

in which $\widetilde{\mathbf{T}}$ is the same BCCB matrix as in Eq. (3.5), and $\tilde{\mathbf{y}} \in \mathbb{R}^{m'n'}$ represents the observed image $\mathbf{y}$ surrounded by the boundary region of width b (see Fig. 3.1). In the proposed framework, this boundary region is estimated, instead of being masked out, as happened in AM. We present two implementations of this framework: an implementation using an off-the shelf deconvolution method that assumes circular boundary conditions, and an efficient implementation based on ADMM in which not all variables are dualized, and for which we present a convergence theorem. We postpone the discussion of the latter implementation to Section 3.3.

Deconvolution methods that do not impose boundary conditions, such as AM and the method proposed by us, are often referred to as methods that use *unknown boundaries*. We use that nomenclature in this work.

As previously mentioned, the framework that we propose is based on model (3.6). From here on, we express the extended blurred image $\tilde{\mathbf{y}}$ in a form that is more convenient for the treatment that follows, and that encompasses not only the case of unknown boundaries, but also all the other cases of unobserved pixels. We denote the number of observed pixels of the blurred image by p, and the number of unobserved pixels (including the above-mentioned boundary zone) by d. Let $\mathbf{z} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^p$ be column vectors containing, respectively, the elements of $\tilde{\mathbf{y}}$ that correspond to unobserved pixels and those that correspond to observed pixels. In a simple deconvolution problem with unknown boundaries, $\mathbf{z}$ contains the boundary zone, and $\mathbf{y}$ contains the observed blurred image. In a combined deconvolution and inpainting problem, $\mathbf{z}$ contains both the boundary zone and the additional unobserved pixels, and $\mathbf{y}$ contains the pixels of the blurred image that were actually observed. We reorder the elements of the extended image $\tilde{\mathbf{y}}$ as $\left[\begin{smallmatrix}\mathbf{y}\\ \mathbf{z}\end{smallmatrix}\right] = \mathbf{P}\tilde{\mathbf{y}}$, where $\mathbf{P}$ is an appropriate permutation matrix, so that the observed pixels are in the first positions and the unobserved pixels are in the last positions of the vector $\left[\begin{smallmatrix}\mathbf{y}\\ \mathbf{z}\end{smallmatrix}\right]$.

Conceptually, the proposed framework is rather simple. It consists of using the blurring model (3.6), and alternately estimating $\mathbf{x}$ and $\mathbf{z}$, as shown in Fig. 3.4. In this framework, step 2, which estimates $\mathbf{x}$, can be performed, essentially, with any existing deblurring method that assumes circular boundary conditions.[2] Step 3 is performed by just computing the reconstructed blurred image, given by $\widetilde{\mathbf{T}}\mathbf{x}$, and selecting from it the pixels that correspond to $\mathbf{z}$.

An important difference of the proposed framework relative to most published deblurring methods, including AM, is that, in this framework, the unobserved pixels of the blurred image (represented by $\mathbf{z}$) are explicitly estimated. This means that we have one more variable to estimate ($\mathbf{z}$).

[2] The framework can also be used with methods that use other boundary conditions. The only difference is in the structure of the matrix $\widetilde{\mathbf{T}}$ used in model (3.6). Instead of being a BCCB matrix, it has the proper structure for the boundary conditions under consideration.

Framework 1

```
1. Initialize $\mathbf{x}^1$ and $\mathbf{z}^1$. Let $k = 1$.
repeat
    2. Compute $\mathbf{x}^{k+1}$ given $\mathbf{x}^k$, $\mathbf{z}^k$.
    3. Compute $\mathbf{z}^{k+1}$ given $\mathbf{x}^{k+1}$.
    4. Increment $k$.
until stopping criterion is satisfied.
```

Figure 3.4.: The proposed deblurring framework.

As given in Fig. 3.4, the proposed framework is rather general. It can be used to design new deblurring methods, an example of which is the efficient ADMM-based method that we propose in Section 3.3. It is also an efficient, high-quality alternative to methods such as edge tapering, to convert existing deblurring methods that impose specific boundary conditions into methods that work with unknown boundaries. We illustrate this in Subsection 3.2.1, by using the proposed framework to convert an off-the-shelf, state-of-the-art deblurring method that assumes circular boundary conditions—IDD-BM3D [93]—into a method that uses unknown boundaries.

3.2.1. Selected experimental results

This subsection illustrates the use of the framework of Fig. 3.4 with a state-of-the-art deblurring method, which assumes periodic boundary conditions, by adapting it to the unknown-boundaries situation. We plugged into the mentioned framework, without modification, a fast, high-quality FFT-based deconvolution method: IDD-BM3D [93], which assumes circular boundary conditions to be able to perform fast matrix operations in the frequency domain by means of the FFT. It takes a frame-based approach to the deconvolution problem, and performs both a deconvolution and a denoising step. Additionally, it runs another deconvolution method [94] for initialization. We used the published IDD-BM3D software, without change, to implement step 2 of Framework 1. As a result of the incorporation into the framework, we obtained a deconvolution method that uses unknown boundaries, and that still retains the speed of FFT-based matrix operations. We compared the results of the resulting method with those of the commonly adopted solution of using edge tapering to deal with the unknown boundaries.

In the experimental tests, we compared three different situations: (a) direct application of IDD-BM3D without the use of Framework 1, to assess the effect of the method's assumption of circular boundary conditions on an image that did not obey those conditions; (b) similar to (a), but preprocessing the observed image by edge tapering, to reduce the effect of the mentioned assumption; and (c) the use of IDD-BM3D within Framework 1, as described above. We ran the experiment using the *cameraman* image with size 256 × 256 pixels, blurred with a 9 × 9 boxcar filter, and with additive i.i.d. Gaussian noise with a blurred-signal-to-noise ratio (BSNR) of 40 dB. See [9, Section III.A] for more implementation details.

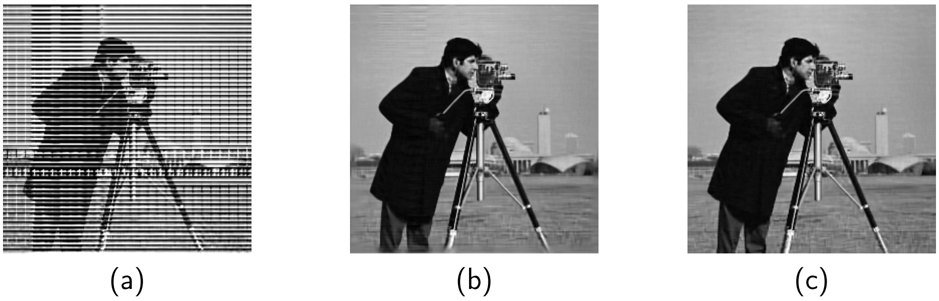


Figure 3.5.: Use of the IDD-BM3D deblurring method and Framework 1 in a blurred image with unknown boundaries. (a) Plain IDD-BM3D (ISNR = -14.62 dB). (b) IDD-BM3D with pre-smoothing of the blurred image's borders (ISNR = 8.27 dB). (c) Framework 1, with step 2 implemented through IDD-BM3D (ISNR = 9.64 dB).

The results of the tests, along with the corresponding improvement in SNR (ISNR) values, are shown in Fig. 3.5. As can be seen, using IDD-BM3D without taking into account that the boundary conditions were not circular [situation (a)] produced very strong artifacts. With edge tapering [situation (b)], the artifacts were much reduced, although some remained visible; there also was some loss of detail near the image borders. With the use of IDD-BM3D within Framework 1 [situation (c)], there were barely any artifacts, and the image remained sharp all the way to the borders. The values of the ISNR agree with these observations.

The example of the use of IDD-BM3D within the proposed framework illustrates the fact that this framework can be used to convert existing deblurring methods that assume artificial boundary conditions to methods that use unknown boundaries. As far as we know, there is no other published way to accomplish this. This is a simple alternative to the use of edge tapering, yielding results of better quality.

3.3. "Partial" ADMM

We now introduce the ADMM-based implementation of the proposed framework. We use the blurring model of Eq. (3.6); the noise $\mathbf{n}$ is assumed to be i.i.d. Gaussian. We use a maximum-*a-posteriori* (MAP) formulation,[3] and consequently, the data-fitting term of our objective function is given by

$$f(\mathbf{x},\mathbf{z}) = \frac{1}{2}\left\| \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} - \mathbf{H}\mathbf{x} \right\|_2^2, \tag{3.7}$$

with $\mathbf{H} = \mathbf{P}\widetilde{\mathbf{T}}$ and $\mathbf{x} \in \mathbb{R}^{p+d}$. The problem to be solved is expressed as

$$\underset{\mathbf{x},\mathbf{z}}{\text{minimize}} \quad f(\mathbf{x},\mathbf{z}) + \phi(\mathbf{D}\mathbf{x}), \tag{3.8}$$

[3] See Footnote 8 of Chapter 1 for details.

where $\mathbf{D} \in \mathbb{R}^{l\times(p+d)}$ is a matrix that extracts a linear representation of the estimated image, such as edges, $l \in \mathbb{N}$ is the number of components of that representation, and $\phi(\mathbf{Dx})$ is a regularizer that promotes some desirable characteristic of images, such as sharp edges.

We start by considering the use of ADMM in its standard form to solve Problem (3.8). The resulting method is not very efficient, because it involves a step that is computationally heavy, but it is useful to motivate the method that we propose, and to analyze some of its properties. We then describe our proposed method, which avoids the abovementioned computational inefficiency.

In what follows, we make use of the variables $\mathbf{v}, \mathbf{d} \in \mathbb{R}^{l+d}$, decomposed as $\mathbf{v} = \left[\begin{smallmatrix}\mathbf{v}_x\\ \mathbf{v}_z\end{smallmatrix}\right]$ and $\mathbf{d} = \left[\begin{smallmatrix}\mathbf{d}_x\\ \mathbf{d}_z\end{smallmatrix}\right]$, with $\mathbf{v}_x, \mathbf{d}_x \in \mathbb{R}^l$ and $\mathbf{v}_z, \mathbf{d}_z \in \mathbb{R}^d$. To apply ADMM to Problem (3.8), we first define

$$\mathbf{u} \triangleq \begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}, \quad \mathbf{K} \triangleq \begin{bmatrix}\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_d\end{bmatrix}, \quad \bar{f}(\mathbf{u}) \quad \triangleq f(\mathbf{x}, \mathbf{z}),$$

and also define

$$\psi(\mathbf{v}) \triangleq \phi(\mathbf{v}_x),$$

so that $\psi(\mathbf{Ku}) = \phi(\mathbf{Dx})$.

We rewrite Problem (3.8) as

$$\begin{aligned} \underset{\mathbf{u},\mathbf{v}}{\text{minimize}} \quad & \bar{f}(\mathbf{u}) + \psi(\mathbf{v}) \\ \text{subject to} \quad & \mathbf{v} = \mathbf{Ku}. \end{aligned} \tag{3.9}$$

By applying ADMM to this problem, we obtain the algorithm

Algorithm 14: Standard ADMM.

```
1 Choose $\mathbf{v}^0 \in \mathbb{R}^{l+d}$, $\mathbf{d}^0 \in \mathbb{R}^{l+d}$;
2 Make $k \leftarrow 1$;
3 while stopping criterion is not satisfied do
4     Choose $\gamma^k > 0$;
5     $\begin{bmatrix}\mathbf{x}^{k+1}\\ \mathbf{z}^{k+1}\end{bmatrix} \in \arg\min_{\mathbf{x},\mathbf{z}} \; f(\mathbf{x},\mathbf{z}) + \frac{\gamma^k}{2}\left\|\mathbf{v}^k - \mathbf{K}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix} - \mathbf{d}^k\right\|^2$;
6     $\mathbf{v}^{k+1} \in \arg\min_{\mathbf{v}} \; \psi(\mathbf{v}) + \frac{\gamma^k}{2}\left\|\mathbf{v} - \mathbf{K}\begin{bmatrix}\mathbf{x}^{k+1}\\ \mathbf{z}^{k+1}\end{bmatrix} - \mathbf{d}^k\right\|^2$;
7     $\mathbf{d}^{k+1} = \mathbf{d}^k - \left(\mathbf{v}^{k+1} - \mathbf{K}\begin{bmatrix}\mathbf{x}^{k+1}\\ \mathbf{z}^{k+1}\end{bmatrix}\right)$;
8     $k \leftarrow k+1$;
9 end
```

which we refer to as *standard ADMM.* As mentioned above, the standard ADMM is not normally computationally efficient. This is due to the fact that, in Line 5 of Algorithm 14, $\mathbf{x}$ and $\mathbf{z}$ need to be estimated simultaneously, and this normally involves the inversion of a large matrix that is not easily diagonalizable. To motivate the solution that we propose, we note that if, in Problem (3.8), we consider minimizing relative to $\mathbf{x}$ and to $\mathbf{z}$ separately, only the minimization relative to $\mathbf{x}$ is difficult to perform. The minimization relative to $\mathbf{z}$ is easy to implement in a computationally efficient way, because it is the minimization of a quadratic function, and the matrix $\mathbf{H} = \mathbf{P}\widetilde{\mathbf{T}}$ is diagonalizable in the frequency domain (with an appropriate permutation, corresponding to the product by $\mathbf{P}$). In view of this, we separate the minimization relative to $\mathbf{x}$ from the minimization relative to $\mathbf{z}$, applying them in an alternating manner, and we apply the ADMM machinery only to the variable $\mathbf{x}$, instead of applying it to $\left[\begin{smallmatrix}\mathbf{x}\\ \mathbf{z}\end{smallmatrix}\right]$, as happened in the standard ADMM. Of course, the convergence guarantees of the standard ADMM do not apply to the proposed method. We, therefore, presented a convergence proof for it (see Corollary 3.3.2, below).

Since we are applying the ADMM machinery only to $\mathbf{x}$, Line 5 of Algorithm 14 is replaced by a minimization of

$$f(\mathbf{x},\mathbf{z}) + \frac{\mu}{2}\left\|\mathbf{v}_x^i - \mathbf{D}\mathbf{x} - \mathbf{d}_x^i\right\|^2, \tag{3.10}$$

which we solve approximately by means of an alternating minimization on $\mathbf{x}$ and $\mathbf{z}$ through one or more block-Gauss-Seidel (BGS) passes. Furthermore, Lines 6 and 7 have to be modified so as to refer only to $\mathbf{x}$, and not to $\left[\begin{smallmatrix}\mathbf{x}\\ \mathbf{z}\end{smallmatrix}\right]$. If we use just one BGS pass to minimize (3.10), the complete method corresponds to the algorithm

Algorithm 15: "Partial" ADMM.

```
1  Choose $\mathbf{v}_x^0 \in \mathbb{R}^l,\ \mathbf{d}_x^0 \in \mathbb{R}^l, \mathbf{v}_z^0 \in \mathbb{R}^d,\ \mathbf{d}_z^0 \in \mathbb{R}^d$;
2  Make $k \leftarrow 1$;
3  while stopping criterion is not satisfied do
4      Choose $\gamma^k > 0$;
5      $\mathbf{x}^{k+1} \in \arg\min_{\mathbf{x}}\ f(\mathbf{x},\mathbf{z}^k) + \frac{\gamma^k}{2}\left\|\mathbf{v}_x^k - \mathbf{D}\mathbf{x} - \mathbf{d}_x^k\right\|^2$;
6      $\mathbf{z}^{k+1} \in \arg\min_{\mathbf{z}}\ f(\mathbf{x}^{k+1},\mathbf{z})$;
7      $\mathbf{v}_x^{k+1} \in \arg\min_{\mathbf{v}_x}\ \phi(\mathbf{v}_x) + \frac{\gamma^k}{2}\left\|\mathbf{v}_x - \mathbf{D}\mathbf{x}^{k+1} - \mathbf{d}_x^k\right\|^2$;
8      $\mathbf{d}_x^{k+1} = \mathbf{d}_x^k - (\mathbf{v}_x^{k+1} - \mathbf{D}\mathbf{x}^{k+1})$;
9      $k \leftarrow k + 1$;
10 end
```

If we use more BGS passes, instead of just one, there is an inner loop consisting of Lines 5 and 6.

As can easily be seen, this method falls within the scope of Framework 1, the main steps being Lines 5 and 6; Lines 7 and 8 are added by the use of the ADMM technique. We call Algorithm 15 (with one or more BGS passes) the *"partial" ADMM*; this designation stems from the fact that we only apply the ADMM technique to $\mathbf{x}$, and not to $\left[\begin{smallmatrix}\mathbf{x}\\ \mathbf{z}\end{smallmatrix}\right]$.

We now address the issue of the convergence of the "partial" ADMM. We start by proving (in Theorem 3.3.1) the convergence of a somewhat more general method, and we then show (in Corollary 3.3.2) that the "partial" ADMM is a special case of that method, and is therefore encompassed by Theorem 3.3.1.

Until now, we have assumed the data-fitting term f to be given by Eq. (3.7). For the proof of convergence, we allow f to have the more general form

$$f(\mathbf{x}, \mathbf{z}) \triangleq \frac{1}{2} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}^T \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} + \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}^T \begin{bmatrix} \mathbf{e} \\ \mathbf{f} \end{bmatrix} + g, \tag{3.11}$$

where $\mathbf{A} \in \mathbb{R}^{(p+d)\times(p+d)}$, $\mathbf{B} \in \mathbb{R}^{(p+d)\times d}$, $\mathbf{C} \in \mathbb{R}^{d\times d}$, $\mathbf{e} \in \mathbb{R}^{p+d}$, $\mathbf{f} \in \mathbb{R}^{d}$, and $g \in \mathbb{R}$, and where we assume that $\mathbf{C}$ is positive definite (PD) and that $\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$ is positive semidefinite (PSD). These assumptions guarantee that f is convex, and are not very restrictive. The set of functions that they encompass is only slightly less general than the set of all convex quadratic functions. To obtain the latter set, the assumption on $\mathbf{C}$ would have to be relaxed to being PSD, but additional assumptions would need to be made (see, e.g., [95, Appendix A.5.5]).

The convergence result is given by the following theorem:

Theorem 3.3.1. *Assume that, in Problem* (3.8) *with f defined by Eq.* (3.11)*, $\mathbf{C}$ is PD, $\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$ is PSD, $\mathbf{D}$ is full column rank, and ϕ is closed proper convex and coercive. Define $\mathbf{K} = \left[\begin{smallmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_d \end{smallmatrix}\right]$. Then, the set of solutions of Problem* (3.8) *is non-empty, the sequence $\left\{\left[\begin{smallmatrix} \mathbf{x}^k \\ \mathbf{z}^k \end{smallmatrix}\right]\right\}$ generated by the "partial" ADMM converges to a solution of that problem, and the sequence $\{\mathbf{v}^p\}$ converges to $\mathbf{K}\left[\begin{smallmatrix} \mathbf{x}^* \\ \mathbf{z}^* \end{smallmatrix}\right]$, where $\left[\begin{smallmatrix} \mathbf{x}^* \\ \mathbf{z}^* \end{smallmatrix}\right]$ is the limit of $\left\{\left[\begin{smallmatrix} \mathbf{x}^k \\ \mathbf{z}^k \end{smallmatrix}\right]\right\}$.*

Proof. For the proofs of this theorem and the following corollary, see [9, Appendix B]. □

Corollary 3.3.2. *For Problem* (3.8) *with f given by Eq.* (3.7)*, Theorem 3.3.1 applies.*

Regarding the practical implementation of the "partial" ADMM, the solutions of the minimization problems that constitute Lines 5 and 6, with f given by (3.7), are given, respectively, by

$$\begin{aligned} \mathbf{x}^{k+1} &= \left[\mathbf{H}^T\mathbf{H} + \mu\mathbf{D}^T\mathbf{D}\right]^{-1} \\ &\qquad \left[\mathbf{H}^T\mathbf{M}_z^T\mathbf{z}^k + \mathbf{H}^T\mathbf{M}_y^T\mathbf{y} + \mu\mathbf{D}^T(\mathbf{v}_x^k - \mathbf{d}_x^k)\right] \\ &= \left[\mathbf{H}^T\mathbf{H} + \mu\mathbf{D}^T\mathbf{D}\right]^{-1} \\ &\quad \left[\mathbf{H}^T \begin{bmatrix} \mathbf{y} \\ \mathbf{z}^k \end{bmatrix} + \mu\mathbf{D}^T(\mathbf{v}_x^k - \mathbf{d}_x^k)\right] \end{aligned} \tag{3.12}$$

and

$$\mathbf{z}^{k+1} = \mathbf{M}_z\mathbf{H}\mathbf{x}^{k+1}, \tag{3.13}$$

where $\mathbf{M}_y = [\mathbf{I}_p\ \mathbf{0}]$ is of size $p \times (p+d)$ and $\mathbf{M}_z = [\mathbf{0}\ \mathbf{I}_d]$ is of size $d \times (p+d)$. Matrices $\mathbf{M}_y$ and $\mathbf{M}_z$ are masking matrices and, in particular, $\mathbf{M}_y$ is equivalent to the masking matrix $\mathbf{M}$ used in the AM method (3.5), i.e., $\mathbf{M} = \mathbf{M}_y\mathbf{P}$.

In practice, we have found it useful to use over-relaxation with a coefficient of 2 in the update of $\mathbf{z}$, and therefore we replace Eq. (3.13) with

$$\mathbf{z}^{k+1} = 2\mathbf{M}_z\mathbf{H}\mathbf{x}^{k+1} - \mathbf{z}^k. \tag{3.14}$$

In (3.12), $\mathbf{H}^T\mathbf{H} = \widetilde{\mathbf{T}}^T\mathbf{P}^T\mathbf{P}\widetilde{\mathbf{T}} = \widetilde{\mathbf{T}}^T\widetilde{\mathbf{T}}$ is a BCCB matrix. If $\mathbf{D}^T\mathbf{D}$ is also BCCB, the matrix inverse in (3.12) can be efficiently computed by means of the FFT. On the other hand, since $\mathbf{H} = \mathbf{P}\widetilde{\mathbf{T}}$, products by $\mathbf{H}$ or $\mathbf{H}^T$ can be computed as products by the BCCB matrix $\widetilde{\mathbf{T}}$ followed or preceded by the appropriate permutation, and therefore can also be efficiently computed by means of the FFT. Consequently, the iterations of the proposed method are computationally efficient, having complexity $O[(p+d)\log(p+d)]$.

3.3.1. Selected experimental results

In this subsection, we discuss some of results presented in Simões et al. [9] regarding the "partial" ADMM discussed before. We compared it with two published state-of-the art methods. The first of these is AM, which was chosen because it is an ADMM-based method specifically developed for the problem of deblurring with unobserved pixels, and therefore bears some resemblance to the proposed partial ADMM. The second method used for comparison was the primal–dual algorithm of Condat (cf. Algorithm 8), which we shall denote by CM. We chose this method because it can be expressed in a form that does not require the inversion of matrices related to the blurring operator, and this inversion is a computational bottleneck of most deblurring methods based on exact blurring models. For completeness, we also show results obtained with an approximation of the standard ADMM (cf. Algorithm 14). As discussed in the previous section, the direct application of that method is impracticable, even for small images, in most present-day computers. We give the results obtained by approximately solving the problem in Line 5 through the conjugate-gradient method. We denote this method by 'ADMM-CG'.

As in [89, 90, 96], we used isotropic total variation (TV) [cf. Eq. (1.13) with $p = 2$] to regularize our problem, i.e., we made $\phi(\mathbf{D}\mathbf{x}) = \lambda\sum_{j=1}^{p+d}\sqrt{(\mathbf{D}_h\mathbf{x})_j^2 + (\mathbf{D}_v\mathbf{x})_j^2}$, with $\mathbf{D} = [\mathbf{D}_h^T\mathbf{D}_v^T]^T$. With this regularizer, we can guarantee the uniqueness of the solution of Problem (3.8) if $\mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{D}) = \{\mathbf{0}\}$, which is true if $\mathbf{1}_p \notin \mathcal{N}(\mathbf{H})$, where $\mathcal{N}(\cdot)$ denotes the null space of a matrix. The latter condition is normally verified, since real-world blur kernels usually have nonzero direct-current (DC) gain.

The guarantee of convergence given by Theorem 3.3.1 requires that matrix $\mathbf{D}$ be full column rank. An assumption of this kind is common in the literature, when studying the convergence of primal–dual methods, even though it can be relaxed in some cases (see, e.g., [91] for the ADMM case). With the isotropic TV regularizer, $\mathbf{D}$ is not full column rank (its rank is $p+d-1$), which means that we did not have a formal guarantee of convergence of the proposed method in our experimental tests. This was not the case for AM, where the use of an extra variable splitting makes $\mathbf{D}$ full rank. It was also not the case for CM, in which $\mathbf{D}$ does not need to be full rank. In practice, we found that the four methods always converged, and we had some experimental evidence that they all converged to the same solution, as reported ahead. If we wished to have a formal guarantee of convergence of the proposed method, we could have used one of two approaches: (a) to use a new variable to decouple the convolution operator from $\mathbf{x}$, which would lead to, e.g., $\mathbf{D} = [\mathbf{D}_h^T \mathbf{D}_v^T \mathbf{H}^T]^T$, which is full column rank if $\mathcal{N}(\mathbf{D}_h) \cap \mathcal{N}(\mathbf{D}_v) \cap \mathcal{N}(\mathbf{H}) = \{\mathbf{0}\}$ (the regularizer would need to be changed accordingly), or (b) to modify the discrete difference matrices, making them full rank, e.g., by adding to them $\varepsilon \mathbf{I}_{p+d}$ with a small $\varepsilon > 0$.

In the proposed method, the complexity of an iteration was dominated by FFTs: three for Lines 5 and 6, and one for Line 7; the number of FFTs thus depended on the number of BGS passes (each pass involved three FFTs). The other two methods both involved four FFTs per iteration.[4] Our method required the storage of one variable with dimension d and five with dimension $p + d$. AM required the storage of seven variables with dimension $p + d$, and CM required the storage of four with dimension $p + d$.

As already mentioned, another parameter of our method is the number of BGS passes. The proof in [9, Appendix B] assumes that a single pass is used, and is easily extendable to any fixed number of passes. If the number of passes changes along the iterations, the proof still applies if that number becomes fixed after a certain number of iterations. For setting this parameter, we found that a simple strategy was useful: for a blur of size $(2b + 1) \times (2b + 1)$, we set the number of passes to b. We found this strategy to yield a good choice of the number of passes, irrespective of the noise level of the observed image. In what follows, we report the results obtained using this adaptive strategy (designated by 'Proposed-AD'), and also the results obtained with just one BGS pass (designated by Proposed-1'); the latter are reported because they correspond to an especially simple, and therefore interesting situation. For more details on the implementation, parameter tuning, and convergence guarantees, see [9, Section III.B].

To check whether the four methods converged to the same result, we ran them for a very large number of iterations (10^6) in a few of the cases mentioned ahead, and found that, in each case, the results were essentially the same for all methods: the root-mean-squared error (RMSE)[5] among the results from different methods were always below 10^{-7}, which corresponds to images that are visually indistinguishable. Given this, we arbitrarily chose, for all tests, the results of one of the methods (AM), after the mentioned 10^6 iterations, as representatives of the solutions of the corresponding problems. We used these representatives as reference images for the evaluation of the quality of the results of the four methods. Our choice of AM to compute the reference images did not especially benefit this method relative to the other ones, because, if we had chosen any of the other two methods instead of AM, the RMSE values of the results of the tests, computed relative to them, would have been essentially the same. We did not use the original sharp images as references, for two reasons. First, given the experiments mentioned in the beginning of this paragraph, we had good reasons to believe that the

[4] The publicly available implementation of AM requires seven FFTs per iteration.

[5] The RMSE was defined as

$$\text{RMSE} = \sqrt{\frac{1}{m' \times n'} \sum_{i=1}^{m' \times n'} (x_i - x_i^r)^2}$$

where x_i and x_i^r denote, respectively, the pixels of the estimated image and of the reference image.

four methods converged to the same fixed point, in each problem. Second, the solutions of the optimization problems were slightly different from the original sharp images, due to the presence of the noise and of the regularizer, and our main interest was in assessing the speed of convergence of the methods to the solutions of the optimization problems, not in the recovery of the original images. If we had used the original images as references, it might happen that, in their path to convergence, some of the methods would pass closer to those images than other methods, and this could make them stop earlier. This would give them an apparently better speed, while they might not truly have a better convergence speed. In what follows, when we mention RMSE values, these were computed relative to the above-mentioned reference images, using the whole images, including the boundary zone, in the computation.

We used, in the deblurring tests, two well-known images: *Lena* (in grayscale) and *cameraman*, both of size 256×256 pixels. These images were blurred with boxcar filters of sizes between 3×3 and 21×21, and with truncated Gaussian filters with supports of the same sizes; for a Gaussian filter with a support of size $l \times l$, we used a standard deviation of $\sqrt{l}$. Fig. 3.6 illustrates the behavior of the four methods during the optimization for one of these tests (*cameraman* image with a 13×13 boxcar blurring filter). The estimated images from all the methods were visually indistinguishable from one another. Figure 3.7 shows a result of the Proposed-AD method. For more results, see [9, Section III.B and Appendix C]. In summary, the Proposed-AD method was faster than AM for small and medium-sized boxcar blurs and for small Gaussian blurs, the two methods had similar speeds for large boxcar blurs and medium-sized Gaussian blurs, and AM was faster for large Gaussian blurs. For the application of the proposed method to the problems of image deblurring with inpainting, image superresolution, and image demosaicing, see [9, Section III.B].

The proposed partial ADMM implementation of Framework 1 (more specifically, the Proposed-AD form) showed a performance similar to the performance of the state-of-the-art AM method, both in terms of final results and of computational speed. CM yielded similar deblurred images, but showed a significantly lower computational performance. We conjecture that this was due to the fact that both the proposed method and AM, being based on ADMM, make use of second-order information on the objective function (in terms of the matrix $\mathbf{H}^T\mathbf{H} + \mu\mathbf{D}^T\mathbf{D}$). CM, on the other hand, does not use any information of this kind. This difference in speed agrees with our experience in other image enhancement problems, in which we have repeatedly found CM-like methods to be significantly slower than ADMM-based ones. For further discussion of the use of second-order information in ADMM, see Section 5.3.

3.4. Dissemination and addendum

The proposed framework and ADMM-based method are detailed in a journal paper [9]. Additionally, a MATLAB impementation of the ADMM-based method is available at `https://github.com/alfaiate/DeconvolutionIncompleteObs`.

For the specific case of superresolution, two methods were recently proposed by Wei

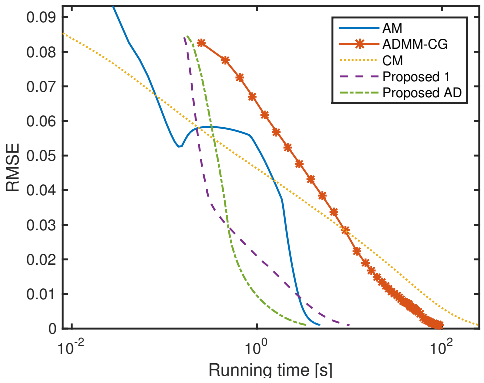
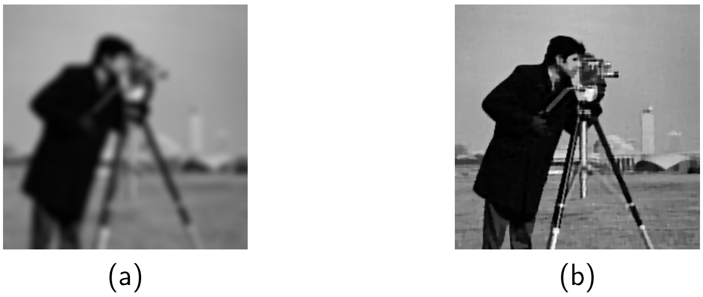


Figure 3.6.: Deblurring: RMSE of the estimated images as a function of running time, for the various methods. This test used the *cameraman* image with a 13 × 13 boxcar blurring filter.


(a)

(b)

Figure 3.7.: Deblurring: observed (a) and estimated (b) images using the Proposed-AD method. The experiments were run for the *cameraman* image with a 13 × 13 boxcar blur.

et. al [97, 98] and by Chan et al. [99]. They are able to efficiently deal with the fact that matrix **T** in model (3.1) is a BTTB matrix, and do not require the use of approximate models.

3.5. Conclusions

We discussed a framework used for deblurring images with unobserved pixels proposed by us in Simões et al. [9]. This framework can be used to convert most deblurring methods to unknown boundaries, irrespective of the specific boundary conditions that those methods assume, being a simple, high-quality alternative to the use of edge tapering. Additionally, we discussed an ADMM-based deblurring method that falls within the mentioned framework, whose proof of convergence was provided in [9]. This method can be seen as a partial ADMM with a non-dualized variable. Experimental results on problems of deconvolution, inpainting, superresolution and demosaicing, with unknown boundaries, showed, for the proposed method, a performance at the level of the state of the art.

4. Hyperspectral superresolution, data fusion, and pansharpening

This chapter describes the contributions of the writer in the following works: Simões et al. [5], Vivone et al. [4], and Veganzones et al. [8]. We start by briefly describing the problems of hyperspectral superresolution, data fusion, pansharpening, and low-dimensional-space identification. We then proceed to present the contributions of this work.

Contents

4.1. Introduction

There are two approaches that can be followed when superresolving hyperspectral images (HSIs). The first one addresses this problem by only processing the observed image (this is the approach discussed in Section 3.1). The second one takes advantage of additional sources of information from the same scene (if they exist), such as panchromatic or multispectral images. HSIs can be combined with these images to produce images with high spatial and spectral resolution. The problem of combining the different images constitutes a *data-fusion* (or sensor-fusion) problem. In this chapter, we focus on the second approach, which can be addressed within the framework of inverse problems.

We first consider the problem of fusing HSIs and multispectral images (MSIs), since the problem of fusing HSIs and panchromatic images (PANs) is a particular case of the former. Consider the use of the notation introduced in Section 1.4, and let the matrix $\mathbf{Z} \in \mathbb{R}^{L_h \times n_m}$ denote the image with high spatial and spectral resolution that is to be estimated (in this case, $\mathbf{Z}$ combines the high spatial resolution of MSIs with the high spectral resolution of HSIs). The hyperspectral ($\mathbf{Y}_h$) and multispectral ($\mathbf{Y}_m$) measurements are typically modeled, respectively, as

$$\begin{aligned} \mathbf{Y}_h &= \mathbf{ZBM} + \mathbf{N}_h, \\ \mathbf{Y}_m &= \mathbf{RZ} + \mathbf{N}_m, \end{aligned} \tag{4.1}$$

where the matrix $\mathbf{B} \in \mathbb{R}^{n_m \times n_m}$ is a spatial-blurring matrix representing the hyperspectral sensor's PSF in the spatial resolution of $\mathbf{Z}$ (the PSF is assumed to be channel-independent), the matrix $\mathbf{M} \in \mathbb{R}^{n_m \times n_h}$ accounts for a uniform subsampling of the hyperspectral image, and is used to yield the lower spatial resolution of the HSI—the columns of $\mathbf{M}$ are a subset of the columns of the identity matrix—, $\mathbf{R} \in \mathbb{R}^{L_m \times L_h}$ is a matrix that accounts for the low spectral resolution of the MSI, and holds in its rows the spectral responses of the multispectral instrument (one per multispectral channel), and $\mathbf{N}_h \in \mathbb{R}^{L_h \times n_h}$ and $\mathbf{N}_m \in \mathbb{R}^{L_m \times n_m}$ represent i.i.d. noise.

This model can also address the problem of fusing HSIs and PANs by considering that a panchromatic image can be seen as a single-channel MSI. By making $L_m = 1$ in Eqs. (4.1), we can specialize $\mathbf{Y}_m$ to the single-channel case. Let $\mathbf{y}_p \in \mathbb{R}^{n_p}$ be a vector denoting this single-channel MSI. We can use it to rewrite the bottom equation in (4.1) as

$$\mathbf{y}_p = \mathbf{r}^T \mathbf{Z} + \mathbf{n}_p, \tag{4.2}$$

where $\mathbf{r} \in \mathbb{R}^{L_h}$ and $\mathbf{n}_p \in \mathbb{R}^{n_p}$. Typically, $n_p > n_m > n_h$. The problem of fusing HSIs and PANs is also known as *hyperspectral pansharpening*. Pansharpening, per se, designates the remote-sensing problem concerned with the fusion of MSIs and PANs, and has received considerable attention from the remote-sensing community. For more details on pansharpening, see, e.g., Amro et al. [100] and Vivone et al. [101]. We also give a small overview of the topic in [5, Section V.D.1].

Compared to the problem of pansharpening, both the HSI-MSI- and the HSI-PAN-fusion problems present challenges of their own. Firstly, since HSIs have a very high dimensionality, the number of variables to estimate is much larger. Secondly, this high dimensionality also has an impact on the computational performance of the algorithms devised to address these problems. Finally, the spectral range covered by HSIs is significantly larger than the one covered by MSIs or PANs and, therefore, many spectral bands of HSIs are not included in any band of MSIs or of PANs.

One way to address the problems related to the high dimensionality of HSIs is to reduce the number of variables involved. This can be accomplished through certain techniques, and some of them are used as building blocks of the algorithms discussed in this chapter; in Subsection 4.2.1, we discuss two of these techniques. They are based on the fact that consecutive bands in a hyperspectral image resemble each other to a high degree, and this implies that there is a large correlation between them. Additionally, the number of endmembers in a given scene is typically much lower than the number of bands, i.e., $P \ll L_h$. These two characteristics suggest that the data lies in a manifold whose dimension is lower than the number of channels of $\mathbf{Y}_h$. In this work, we assume that this manifold is linear, i.e., that it is vector space.[1] Under this assumption, the HSI has the following linear representation:

$$\mathbf{Z} = \mathbf{E}\mathbf{X}, \tag{4.3}$$

where $\mathbf{E} = [\mathbf{e}_1, \cdots, \mathbf{e}_{L_s}] \in \mathbb{R}^{L_h \times L_s}$ is a matrix whose columns span the same space as the columns of $\mathbf{Z}$, $L_s \ll L_h$ is the dimension of the low-dimensional space, and $\mathbf{X} \in \mathbb{R}^{L_s \times n_m}$ is a matrix whose elements are the representation coefficients of $\mathbf{Z}$ in the low-dimensional space.

For an overview of the topic of HSI-MSI-fusion algorithms and hyperspectral super-resolution, see [5, Section I]. An addendum to that overview is given in Section 4.5.

4.1.1. Contributions and outline

In Simões et al. [5], we propose a method that is able to infer images that combine the high spectral and high spatial resolutions of hyperspectral images and multispectral images through the minimization of a convex function containing two quadratic data-fitting terms and an edge-preserving regularizer. The method can also be used to address the problem of fusing HSIs and PANs. We also propose a method to estimate the spatial and the spectral operators linked, respectively, with the HSI and the MSI (or PAN) acquisition processes.

In addition to the description of the contributions of Simões et al. [5], made in Section 4.2, this chapter also introduces, in Sections 4.3 and 4.4, the contributions made by the writer in the context of the work of Vivone et al. [4] and of Veganzones et al. [8]. Since the writer was not the main contributor to those works, the description of these contributions given in this work is brief. In Vivone et al. [4], we address an issue present in the pansharpening algorithms based on the so-called *detail-injection model.* These algorithms are usually comprised of two steps: (a) extraction of the spatial details of the panchromatic image, and (b) their subsequent injection into the multisplectral one. By "spatial details", we refer to the spatial characteristics of the PAN that are not observed in the MSI. In order to find these characteristics, one may use a procedure that requires the application of a low-pass filter to the PAN. The response function of this filter is typically an approximation of the modulation-transfer function (MTF) of the multispectral sensor. Current methods to make this approximation can be inadequate in practice, and we develop a new algorithm for estimating a low-pass filter that only makes use of the observed MSI and PAN. In Veganzones et al. [8], we take advantage of the fact that hyperspectral images usually lie in a low-dimensional space to design a method that addresses the problem of fusing hyperspectral and multispectral images. Methods based on this fact perform at the level of the state of the art if the HSIs actually lie in a low-dimensional space. However, if the dimensionality of this space is not low, in the sense that it is larger than the number of multispectral bands, the performance of these methods decreases. We propose a local approach to cope with this difficulty by exploiting

[1] This is a standard assumption in the remote-sensing literature. See Footnote 16 of Chapter 1.

the fact that real-world HSIs are locally low rank since, in a small spatial neighbourhood, the number of different materials is typically small.

The structure of this chapter is as follows. Section 4.2 describes the method proposed by Simões et al. [5]. Subsection 4.2.1 details the data-fusion method, and Subsection 4.2.2 presents the method used to estimate the spatial blur and the spectral linear operators. Sections 4.3 and 4.4 discuss the contributions of the writer in the context of the works of Vivone et al. [4] and of Veganzones et al. [8], respectively. Section 4.5 details where these works were made available to the research community, and lists some related works that were published after ours.

4.2. Hyperspectral image superresolution: A convex formulation

This section introduces the ideas presented in Simões et al. [5]. In that work, the problem of inferring images that combine the high spectral and high spatial resolutions of hyperspectral images and multispectral images, respectively, is formulated as the minimization of a convex function containing two quadratic data-fitting terms and an edge-preserving regularizer. The data-fitting terms are based on model (4.1), and account for blur, different resolutions, and additive noise. The regularizer promotes piecewise-smooth solutions with discontinuities aligned across the hyperspectral bands, and is a form of total-variation regularization that takes into account both the spatial and the spectral characteristics of the data. It is known as vector total variation (VTV) [102]. In order to solve the optimization problem, we use an ADMM-based method called split augmented-Lagrangian shrinkage algorithm (SALSA) [64], and explore the inherent redundancy of the images with data-reduction techniques to formulate the problem in a computationally efficient way. This method, which is termed *HySure*, for Hyperspectral Superresolution, allows one to fuse hyperspectral data with either multispectral or panchromatic images. It outperformed the state of the art at the time of its publication, as illustrated in a series of experiments with simulated and real-life data. Additionally, we also devise a method to estimate the spatial blur $\mathbf{B}$ and the spectral blur $\mathbf{R}$ linear operators.

In what follows, we discuss our two main contributions in that work, namely the data-fusion method and the method to estimate the operators $\mathbf{B}$ and $\mathbf{R}$. We also present some selected experimental results that compare our method with others from the literature.

4.2.1. Data-fusion method

It may not be practical to work directly on the hyperspectral data due to its high dimensionality. Fortunately, as discussed in Section 4.1, a HSI usually has a representation on a low-dimensional space, a fact that may be useful when solving inverse problems involving these images (see below for an explanation). In order to identify this space, we can use domain-specific techniques or standard techniques. Spectral-unmixing algorithms are examples of domain-specific techniques, in the sense that take into account the physical process that gave origin to $\mathbf{Y}_h$. In this case, $\mathbf{E}$ would be the spectral-signature matrix obtained from $\mathbf{Y}_h$, and $\mathbf{X}$ would represent the abundance fractions of the endmembers for every pixel of $\mathbf{Z}$. An example of a standard technique is singular-value decomposition (SVD) [cf. Eq. (1.2)], which can be used to obtain the factorization $\mathbf{Y}_h = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\boldsymbol{\Sigma}$ is a rectangular diagonal matrix containing the singular values, which are assumed to be in non-increasing order. Denote by $\hat{\boldsymbol{\Sigma}}$, $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$, respectively, the truncated matrices obtained by discarding the rows and columns with the smallest singular values from $\boldsymbol{\Sigma}$ and the corresponding columns of $\mathbf{U}$ and $\mathbf{V}$. A low-dimensional approximation of $\mathbf{Y}_h$ is given by $\hat{\mathbf{U}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{V}}^*$. In this approach, we make $\mathbf{E} = \hat{\mathbf{U}}$. Due to the low intrinsic dimensionality of the hyperspectral data, most of the singular values are rather small, allowing a very significant dimensionality reduction while retaining a rather faithful approximation of $\mathbf{Y}_h$. If $\mathbf{N}_h = \mathbf{0}$ and all the discarded singular values are zero, this representation spans the true signal subspace. If the former condition on $\mathbf{N}_h$ is not obeyed but $\mathbf{N}_h$ is i.i.d., this representation corresponds to the maximum likelihood estimate of that subspace. However, if the noise is non-i.i.d., the estimation of the subspace is more complex; see, for example, [103] for details, and for algorithms oriented to subspace estimation in hyperspectral applications.

After identifying the low-dimensional space, the hyperspectral data can than be projected into it. One may prefer to work directly on the projected data in, e.g., an inverse problem involving the HSI, since the number of variables to be estimated is significantly reduced. This has a number of advantages: (a) it is computationally more efficient, (b) the estimates are normally more accurate, and (c) the signal-to-noise ratio (SNR) of the HSI is improved [29, Section III.A]. With any of the two factorizations, we replace the first equation in (4.1) with

$$\mathbf{Y}_h = \mathbf{EXBM} + \mathbf{N}_h, \tag{4.4}$$

where the error due to the dimensionality reduction has been incorporated into $\mathbf{N}_h$.

The problem that we are trying to solve is strongly ill-posed, and therefore needs adequate regularization. The regularizer that we use is given by

$$\varphi(\mathbf{XD}_h, \mathbf{XD}_v) \triangleq \sum_{j=1}^{n_m} \sqrt{\sum_{i=1}^{L_s} \Big\{ \big[(\mathbf{XD}_h)_{ij}\big]^2 + \big[(\mathbf{XD}_v)_{ij}\big]^2 \Big\}}, \tag{4.5}$$

where $(\mathbf{A})_{ij}$ denotes the element in the ith row and jth column of matrix $\mathbf{A}$, and the products by matrices $\mathbf{D}_h$ and $\mathbf{D}_v$ compute the horizontal and vertical discrete differences of an image, respectively, with periodic boundary conditions.[2] This regularizer, which is a form of VTV, imposes sparsity in the distribution of the absolute gradient of an image, meaning that transitions between the pixels of an image should be smooth in the spatial dimension, except for a small number of them, which should coincide with details such as edges. Zhao *et al.* [104] proposed an isotropic TV scheme for hyperspectral image deblurring in a band-by-band manner. This means that each band was regularized independently from the other ones. This approach has a shortcoming: it does not take into account that edges and other details normally have the same locations in most bands.

[2] See Section 3.1 for a definition of these conditions.

The vector form of the regularizer, which we use in our work, promotes solutions in which edges and other details are aligned among the different bands.

We apply the regularizer to the reduced-dimensionality data **X**, and not to **Z** itself. This is indeed reasonable, since the subspace spanned by **E** is the same as the one where **Z** resides (or an approximation, when using truncated SVD), and by regularizing **X** we are indirectly regularizing **Z**.

The optimization problem based on the proposed regularizer is

$$\begin{aligned}\underset{\mathbf{X}}{\text{minimize}} \qquad & \frac{1}{2}\left\|\mathbf{Y}_h - \mathbf{EXBM}\right\|_F^2 + \frac{\lambda_m}{2}\left\|\mathbf{Y}_m - \mathbf{REX}\right\|_F^2 \\ & \quad + \lambda_\varphi \varphi\big(\mathbf{XD}_h, \mathbf{XD}_v\big). \end{aligned} \tag{4.6}$$

The first two terms are data-fitting terms and are used to enforce the model defined in Eq. (4.1). The last term is the regularizer. The parameters λ_m and λ_φ control the relative weights of the various terms.

Problem (4.6) is convex, but is rather hard to solve, due to the nature of the regularizer, which is nonsmooth. Additional difficulties are raised by the large size of **X** and by the presence of the downsampling operator **M** in one of the quadratic terms, preventing a direct use of the FFT in optimizations involving this term. We deal with these difficulties by using SALSA, which is an ADMM-based method. It involves the introduction of auxiliary variables into the optimization problem, through the so-called variable-splitting technique. We split the original optimization variable **X** into a total of five variables: one which we still call **X**, and four auxiliary variables, $\mathbf{V}_1$ to $\mathbf{V}_4$. Problem (4.6) becomes

$$\begin{aligned}\underset{\mathbf{X}}{\text{minimize}} \qquad & \frac{1}{2}\left\|\mathbf{Y}_h - \mathbf{EV}_1\mathbf{M}\right\|_F^2 + \frac{\lambda_m}{2}\left\|\mathbf{Y}_m - \mathbf{REV}_2\right\|_F^2 \\ & \quad + \lambda_\varphi \varphi\big(\mathbf{V}_3, \mathbf{V}_4\big) \\ \text{subject to} \qquad & \mathbf{V}_1 = \mathbf{XB}, \\ & \mathbf{V}_2 = \mathbf{X}, \\ & \mathbf{V}_3 = \mathbf{XD}_h, \\ & \mathbf{V}_4 = \mathbf{XD}_v. \end{aligned} \tag{4.7}$$

For notational simplicity, we define the matrices **V** and **H**,

$$\mathbf{V} \triangleq \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ \mathbf{V}_3^T \\ \mathbf{V}_4^T \end{bmatrix}, \quad \mathbf{H} \triangleq \begin{bmatrix} \mathbf{B}^T \\ \mathrm{Id} \\ \mathbf{D}_h^T \\ \mathbf{D}_v^T \end{bmatrix},$$

and the cost function

$$\begin{aligned} f(\mathbf{V}) \triangleq & \frac{1}{2}\left\|\mathbf{Y}_h - \mathbf{EV}_1\mathbf{M}\right\|_F^2 + \frac{\lambda_m}{2}\left\|\mathbf{Y}_m - \mathbf{REV}_2\right\|_F^2 \\ & + \lambda_\varphi \varphi\big(\mathbf{V}_3, \mathbf{V}_4\big). \end{aligned}$$

We can express Problem (4.7) as

$$\begin{aligned} \underset{\mathbf{X}}{\text{minimize}} \quad & f(\mathbf{V}) \\ \text{subject to} \quad & \mathbf{V} = \mathbf{H}\mathbf{X}^T. \end{aligned} \tag{4.8}$$

This problem has the following augmented Lagrangian

$$\mathcal{L}(\mathbf{X},\mathbf{V},\mathbf{A}) = f(\mathbf{V}) + \frac{\mu}{2}\left\|\mathbf{H}\mathbf{X}^T - \mathbf{V} - \mathbf{A}\right\|_F^2, \tag{4.9}$$

where $\mathbf{A}$ is the so-called scaled dual variable [91], and μ is a positive constant, called penalty parameter. We are now ready to apply the ADMM method, which yields Algorithm 16. As we can see, SALSA solves the original, complex optimization problem through an iteration on a set of much simpler problems. The constraints are taken into account, in an approximate way, by minimizing the augmented Lagrangian of the problem relative to the auxiliary variables.

Algorithm 16: Pseudocode for the HySure algorithm.

```
Require: data: $\mathbf{Y}_h$, $\mathbf{Y}_m$; regularization parameters: $\lambda_m$, $\lambda_\varphi$; penalty parameter:
    $\mu$; matrices $\mathbf{R}$, $\mathbf{B}$ and $\mathbf{E}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    $\mathbf{X}^{k+1} \leftarrow \arg\min_{\mathbf{X}} \; \mathcal{L}\left(\mathbf{X}, \mathbf{V}^k, \mathbf{A}^k\right)$
    $\mathbf{V}^{k+1} \leftarrow \arg\min_{\mathbf{V}} \; \mathcal{L}\left(\mathbf{X}^{k+1}, \mathbf{V}, \mathbf{A}^k\right)$
    $\mathbf{A}^{k+1} \leftarrow \mathbf{A}^k - \left[\mathbf{H}\left(\mathbf{X}^{k+1}\right)^T - \mathbf{V}^{k+1}\right]$
    $k \leftarrow k + 1$;
end
```

The minimization with respect to $\mathbf{X}$ is a quadratic problem with a block-cyclic system matrix, which can be efficiently solved by means of the FFT. Minimizing with respect to the auxiliary variables is done by solving three different problems, whose solutions correspond to three Moreau proximity operators. The minimization with respect to $\mathbf{V}_1$ is a quadratic problem which is efficiently solved via FFTs, and the minimization relative to $\mathbf{V}_2$ is also quadratic; these two problems involve matrix inverses which can be computed in advance. Finally, the minimization with respect to $\mathbf{V}_3$ and $\mathbf{V}_4$ corresponds to a pixel-wise *vector soft-thresholding* operation.

The details of the optimization, as well as an analysis of the algorithm's complexity, are presented in [5, Appendix].

4.2.2. Estimation of the spatial blur and of the spectral response

In the literature on the topic, the HSI-MSI-fusion problem is very often dealt with as a non-blind problem, in the sense that the spatial and spectral responses of the sensors are assumed to be known [105, 106, 107, 108]. In practice, however, the information that is available about these responses is often scarce and/or somewhat inaccurate. We deal with this problem blindly—in the sense that we assume these responses to be unknown—, formulating another convex problem to estimate them by making only minimal assumptions: we assume that the spatial response has limited support and that both responses are relatively smooth.

In this blind setting, the matrices $\mathbf{B}$ and $\mathbf{R}$ are estimated from the observed images. The advantages of doing so are threefold. First, as previously mentioned, the available information about the sensors can be rather scarce. Second, it may be hard to precisely adapt that information to the model that is being used for data fusion. Third, there may be discrepancies between the real spatial and spectral responses and the data supplied by the manufacturers. These can be due to several causes, such as atmospheric conditions, postprocessing artifacts, and even the variability within the observed scene [109]. This problem has been addressed before in the literature. See [5, Section IV] for a review.

From Eqs. 4.1, recall that, without noise,

$$\mathbf{Y}_h = \mathbf{ZBM}, \quad \mathbf{Y}_m = \mathbf{RZ},$$

which implies that

$$\mathbf{RY}_h = \mathbf{Y}_m\mathbf{BM}. \tag{4.10}$$

We assume that matrix $\mathbf{B}$ accounts for a 2-D cyclic convolution, and that the convolution kernel has finite support contained in a square window of size $\sqrt{n_b}$ centered at the origin, thus containing n_b pixels.

Taking Eq. (4.10) into account, we infer $\mathbf{R}$ and $\mathbf{B}$ by solving the optimization problem

$$\underset{\mathbf{B},\mathbf{R}}{\text{minimize}} \quad \left\|\mathbf{RY}_h - \mathbf{Y}_m\mathbf{BM}\right\|_F^2 + \lambda_b\phi_B(\mathbf{B}) + \lambda_R\phi_R(\mathbf{R}), \tag{4.11}$$

where $\phi_B(\cdot)$ and $\phi_R(\cdot)$ are quadratic regularizers that are discussed in detail below, and $\lambda_b, \lambda_R \geq 0$ are the respective regularization parameters. Matrix $\mathbf{B}$, and possibly also matrix $\mathbf{R}$, are subject to some constraints discussed below.

A special consideration needs to be made regarding the estimation of the spectral response. This is due to the fact that, when using the observed data, it is not possible to fully estimate matrix $\mathbf{R}$. The reason for this is that the hyperspectral data normally span only a low-dimensional subspace of the full spectral space. Only the component of $\mathbf{R}$ parallel to that subspace can be estimated. This is not a drawback, however, since the component of $\mathbf{R}$ orthogonal to that subspace has essentially no influence on the result of the image fusion. In fact, if we write $\mathbf{R} = \mathbf{R}_{\|} + \mathbf{R}_{\perp}$, where $\mathbf{R}_{\|} = \mathbf{RP}_{\|}$ and $\mathbf{R}_{\perp} = \mathbf{RP}_{\perp}$, and $\mathbf{P}_{\|}$ and $\mathbf{P}_{\perp}$ denote the projection matrices onto the subspaces spanned by the original hyperspectral vectors and onto the subspace orthogonal to it, respectively, we have $\mathbf{RY}_h = \mathbf{R}_{\|}\mathbf{Y}_h + \mathbf{R}_{\perp}\mathbf{Y}_h = \mathbf{R}_{\|}\mathbf{Y}_h$, since $\mathbf{R}_{\perp}\mathbf{Y}_h$ is zero. For the product $\mathbf{RZ}$, which is involved in the fusion problem, we have $\mathbf{RZ} \approx \mathbf{R}_{\|}\mathbf{Z}$, since $\mathbf{Z}$ spans approximately the same subspace as $\mathbf{Y}_h$, because it corresponds to an image containing the same endmembers.

Let $[\mathbf{Y}_m\mathbf{B}]_{:j}$ denote the jth column of $\mathbf{Y}_m\mathbf{B}$, let $\mathbf{b} \in \mathbb{R}^{n_b}$ denote the columnwise ordering of the convolution kernel, and let $\mathbf{P}_j \in \mathbb{R}^{n_m \times n_b}$ denote a matrix which selects from $\mathbf{Y}_m$ a patch such that

$$[\mathbf{Y}_m\mathbf{B}]_{:j} = (\mathbf{Y}_m\mathbf{P}_j)\mathbf{b}.$$

With these definitions in place, a slight modification of the Problem (4.11) is

$$\begin{aligned} \underset{\mathbf{b},\mathbf{R}}{\text{minimize}} \quad & \sum_{j=1}^{n_h} \Big\| \mathbf{R}\mathbf{Y}_{h,:j} - \mathbf{Y}_{m,j}\mathbf{b} \Big\|_2^2 \\ & \qquad + \lambda_b \phi_b(\mathbf{b}) + \lambda_R \phi_R(\mathbf{R}) \\ \text{subject to} \quad & \mathbf{b}^T\mathbf{1} = 1, \end{aligned} \tag{4.12}$$

where $\mathbf{Y}_{h,:j}$ denotes the jth column of $\mathbf{Y}_h$, $\mathbf{Y}_{m,j} \triangleq \big[(\mathbf{Y}_m\mathbf{P}_{c_j})\big] \in \mathbb{R}^{L_m \times n_b}$, with c_j denoting the column of $\mathbf{Y}_m$ corresponding to the jth column of $\mathbf{Y}_h$, $\phi_b(\mathbf{b}) \triangleq \phi_B(\mathbf{B})$, and the normalization condition $\mathbf{b}^T\mathbf{1} = 1$ imposes unit DC gain of the blur.

We note that Problem (4.12) is a quadratic program with only equality constraints and, therefore, using Lagrange multipliers, its solution can be obtained by solving a linear system of equations. However, even though we have a closed-form solution, because the size of the optimization variables (*i.e.*, $n_b + L_m \times L_h$) is usually of the order of thousands, it may be useful to solve Problem (4.12) via alternated minimization with respect to $\mathbf{b}$ and $\mathbf{R}$.

The optimization with respect to $\mathbf{b}$ leads to the following regularized least-squares problem:

$$\begin{aligned} \underset{\mathbf{b}}{\text{minimize}} \quad & \sum_{j=1}^{n_h} \Big\| \mathbf{R}\mathbf{Y}_{h,:j} - \mathbf{Y}_{m,j}\mathbf{b} \Big\|_2^2 \\ & \qquad + \lambda_b \Big(\big\|\mathbf{D}_h\mathbf{b}\big\|_2^2 + \big\|\mathbf{D}_v\mathbf{b}\big\|_2^2 \Big) \\ \text{subject to} \quad & \mathbf{b}^T\mathbf{1} = 1, \end{aligned} \tag{4.13}$$

The two last terms of the function being minimized in (4.13) correspond to $\phi_b(\cdot)$, which, in this case, is a noise-removing regularizer that smooths the estimated convolution kernel by promoting that the values of the differences between neighboring pixels be small. As before, $\mathbf{D}_h$ and $\mathbf{D}_v$ compute the horizontal and vertical discrete differences of the convolution kernel, with dimensions adjusted for this particular case.

An approximate solution for Problem (4.13) is computed by first relaxing the constraint, estimating the filter without the normalization condition, and then normalizing the result to unit DC gain. The solution of the unconstrained problem is given by

$$\begin{aligned} \mathbf{b}^* = \Big[\sum_{j=1}^{n_h} \mathbf{Y}_{m,j}^T\mathbf{Y}_{m,j} + \lambda_b\big(\mathbf{D}_h^T\mathbf{D}_h + \mathbf{D}_v^T\mathbf{D}_v\big) \Big]^{-1} \\ \Big[\sum_{j=1}^{n_h} \mathbf{Y}_{m,j}^T\mathbf{R}\mathbf{Y}_{h,:j} \Big]. \end{aligned} \tag{4.14}$$

The support covered by $\mathbf{b}$ is specified by the user. We have found, experimentally, that the choice of this support does not have much influence on the blur estimate, as long as it encompasses the support of the actual blur.

Concerning the estimation of $\mathbf{R}$, we use the regularizer $\phi_R(\cdot)$ in order to deal with the indetermination of the orthogonal component, and to reduce estimation noise. In the cases in which there is information about the overlap between bands of the HSI and the MSI, we constrain the elements of $\mathbf{R}$ that correspond to non-overlapping bands to zero.

The estimation of $\mathbf{R}$ can be made independently for each of the MSI bands. Let $\mathbf{r}_i^T$ denote a row vector containing the ith row of $\mathbf{R}$ without the elements that are known to correspond to hyperspectral bands that do not overlap the ith multispectral band, and by $\mathbf{Y}_{h,i}$ denote the matrix $\mathbf{Y}_h$ without the rows corresponding to those same bands. The optimization of (4.12) is decoupled with respect to the rows of $\mathbf{R}$ and may be written as

$$\underset{\mathbf{r}_i}{\text{minimize}} \quad \left\|\mathbf{r}_i^T \mathbf{Y}_{h,i} - \mathbf{Y}_{m,i:}\mathbf{BM}\right\|_2^2 + \lambda_R \left\|\mathbf{D}\mathbf{r}_i\right\|_2^2, \tag{4.15}$$

in which $\mathbf{Y}_{m,i:}$ is the ith row of $\mathbf{Y}_m$, and the product by $\mathbf{D}$ computes the differences between the elements in $\mathbf{r}_i$ corresponding to contiguous hyperspectral bands. The solution of (4.15) is given by

$$\mathbf{r}_i^* = \left[\mathbf{Y}_{h,i}\mathbf{Y}_{h,i}^T + \lambda_R \mathbf{D}^T\mathbf{D}\right]^{-1} \mathbf{Y}_{h,i}\left[\mathbf{Y}_{m,i:}\mathbf{BM}\right]^T. \tag{4.16}$$

The estimation of each of the matrices $\mathbf{B}$ and $\mathbf{R}$, as presented so far, requires the knowledge of the other matrix. In order to estimate both, and instead of using alternating optimization as proposed before, we adopt an even simpler technique. We start by estimating $\mathbf{R}$. To do this without knowing $\mathbf{B}$, we first blur both spectral images with a spatial blur that is much stronger than the one produced by $\mathbf{B}$, so that the effect of $\mathbf{B}$ becomes negligible. This, conveniently, also minimizes the effect of possible misregistration between the hyperspectral and multispectral images. Following this, we estimate the spectral response $\mathbf{R}$ using Eq. (4.16), setting the kernel of the spatial blur between the strongly blurred multispectral and hyperspectral images to a delta impulse. Finally, we estimate the spatial blur $\mathbf{B}$ using Eq. (4.14) on the original (unblurred) images, with the value of $\mathbf{R}$ just found. Fig. 4.1 summarizes the estimation method.

We now discuss the set of solutions of (4.12), which is an important issue in our approach to the estimation of $\mathbf{b}$ and $\mathbf{R}$, closely related to that of identifiability. Given that the objective function is quadratic, a sufficient condition for it to have a unique solution is that its Hessian matrix be PD. Assuming that $\lambda_b, \lambda_R > 0$, the null space associated with the regularization terms is the set

$$\mathbf{A} \triangleq \{(\mathbf{R}, \mathbf{b}) \, : \, \mathbf{R} = \mathbf{c}\mathbf{1}_{L_h}^T, \, \mathbf{b} = d\mathbf{1}_{n_b}, \mathbf{c} \in \mathbb{R}^{L_m}, d \in \mathbb{R}\},$$

where we have assumed that the spectral response of the multispectral channels spans over the entire L_h hyperspectral bands, and these bands are contiguous in frequency. The case in which the spectral response of the multispectral channels spans over subsets of the L_h bands corresponds to a minor modification of the reasoning provided below. The case in which the hyperspectral bands are not contiguous is somewhat more elaborate, but would follow the same line of reasoning.

For any $(\mathbf{R}, \mathbf{b}) \in \mathbf{A}$, we may write

$$\sum_{j=1}^{n_h} \left\| \mathbf{R}\mathbf{Y}_{h,:j} - \mathbf{Y}_{m,j}\mathbf{b} \right\|_2^2 = \sum_{j=1}^{n_h} \left\| y_{h,j}\mathbf{c} - \mathbf{y}_{m,j}d \right\|_2^2, \tag{4.17}$$

for some $\mathbf{c} \in \mathbb{R}^{L_m}, d \in \mathbb{R}$ and where $y_{h,j} \triangleq \mathbf{1}_{L_h}^T \mathbf{Y}_{h,:j}$ and $\mathbf{y}_{m,j} \triangleq \mathbf{Y}_{m,j}\mathbf{1}_{n_b}$. Let us suppose that there exits a nonzero couple $(\mathbf{c}, d)$ nulling all the n_h quadratic terms in the right hand side of (4.17). In this case, all vectors $\mathbf{y}_{m,j}$, for $j = 1, \ldots, n_h$ would be collinear with $\mathbf{c}$. Having into consideration that the components of $\mathbf{y}_{m,j}$ represent the average intensities in the L_m multispectral bands in the patch $\mathbf{P}_{c_j}$, such a scenario is highly unlikely, implying that the intersection of the subspace $\mathbf{A}$ with the null space associated with the data term shown in the left hand side of (4.17) is empty, except for the origin. We conclude, therefore, that the Hessian of the quadratic objective function present in (4.12) is positive definite and, thus, the solution of the corresponding optimization exists and is unique. An important consequence of this uniqueness is that the subproblems (4.13) and (4.15) have unique solutions; moreover, the system matrices present in the Eqs. (4.14) and (4.16) are nonsingular.

```
Require: data: $\mathbf{Y}_h$ and $\mathbf{Y}_m$; regularization parameters: $\lambda_R$ and $\lambda_B$.
  Blur $\mathbf{Y}_m$ with a strong blur.
  Blur $\mathbf{Y}_h$ with a correspondingly scaled blur.
  Estimate $\mathbf{R}$ using Eq. (4.16) on the blurred data.
  Estimate $\mathbf{B}$ using Eq. (4.14) on the original observed data.
  Normalize $\mathbf{b}$ to unit DC gain.
```

Figure 4.1.: Summary of the method to estimate the spectral response $\mathbf{R}$ and the spatial blur; note that $\mathbf{B}$ and $\mathbf{b}$ are just two different ways of expressing this spatial blur. $\mathbf{Y}_m$ and $\mathbf{Y}_h$ refer to the multispectral and hyperspectral observations, respectively.

4.2.3. Selected experimental results

In this subsection, we discuss some of the results presented in Simões et al. [5]. For more experimental results, a brief description of the methods used in the comparisons with our fusion method, details on the implementation of the algorithm, and a description of the indices that were used to evaluate the quality of the results, see [5, Section V].

One of the datasets that was used to test the different algorithms consisted of images taken above Paris (see Fig. 4.2a), and was obtained by two instruments on board the Earth Observing-1 Mission (EO-1) satellite, the Hyperion instrument and the Advanced Land Imager (ALI). Hyperion is a hyperspectral imager with a spatial resolution of

30 meters; the ALI instrument provides both multispectral and panchromatic images at resolutions of 30 and 10 meters, respectively [110].[3] The hyperspectral and panchromatic images were directly used for experiments on HSI-PAN fusion, and therefore we had no access to the ground truth. We hence only show false color representations of the estimated images, for visual inspection. Fig. 4.2 shows the results. For experiments on the fusion of hyperspectral and multispectral images, we needed the HSI to have lower resolution than the MSI, and therefore we first reduced the spatial resolution of the hyperspectral image by blurring it with a low-pass filter and downsampling. The original hyperspectral image, before blurring and downsampling, was used as ground truth. The results of these tests are shown in Fig. 4.3. Fig. 4.4a shows the RMSE between the ground truth and the images estimated by both methods for each pixel (a $\mathbb{R}^{L_h}$ vector), with the results sorted in ascending order of error. The proposed method surpassed the other one in all tests.

4.3. Pansharpening based on semiblind deconvolution

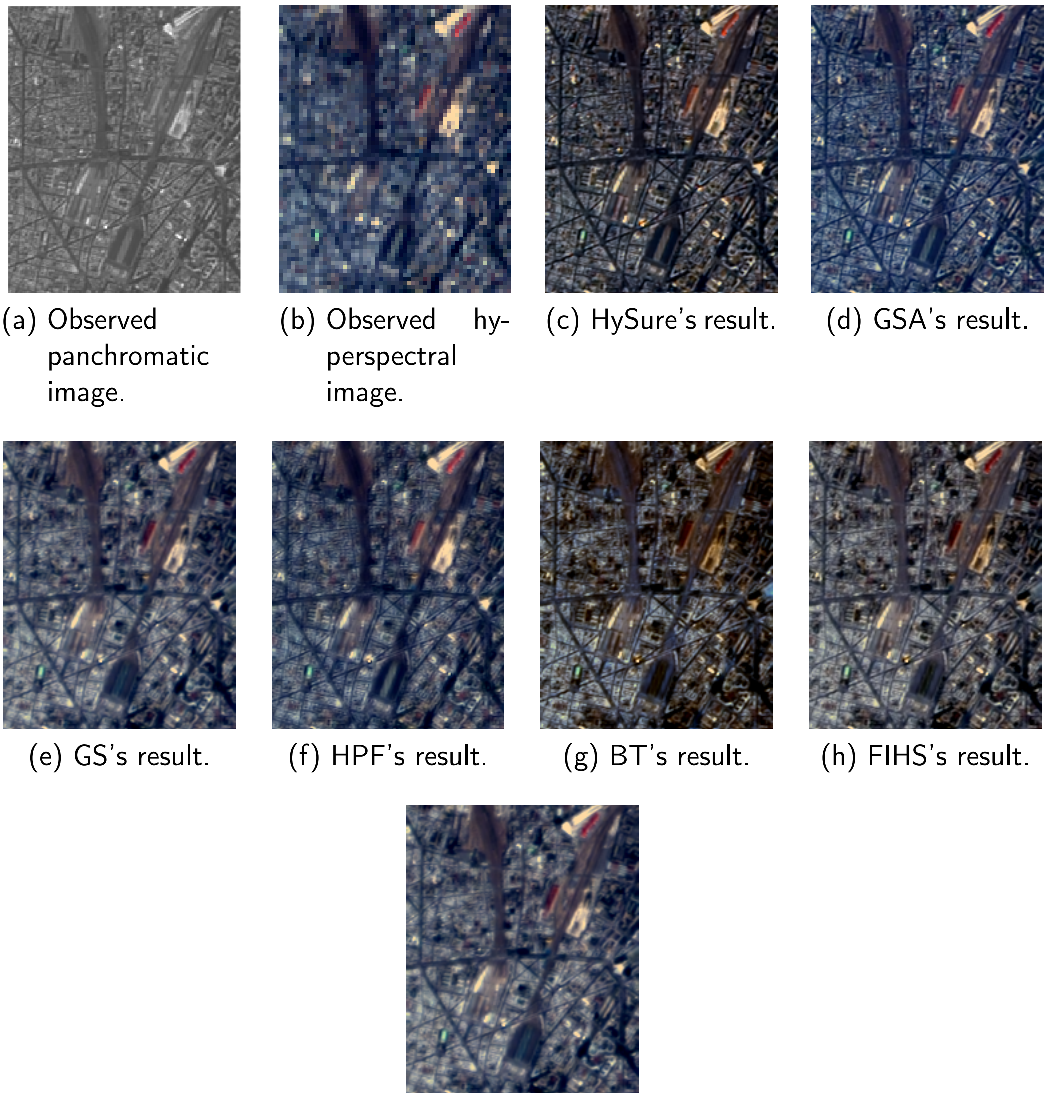
This section describes the contribution made by the writer in the context of the work of Vivone et al. [4]. We follow the formulation of the pansharpening problem described in Section 4.1. A number of pansharpening algorithms require one to produce a version of the panchromatic image with the same level of spatial detail as of the multispectral one. This is accomplished by filtering the observed PAN with an appropriate low-pass filter. Typically, this is a Gaussian filter whose response function is made to match the response function of the multispectral sensor (which is measured before launch) as closely as possible. The algorithms based on a detail-injection model then use this filtered PAN to produce another image containing only the spatial details that are absent from the MSI. They do this by subtracting the filtered PAN from the original panchromatic image. Subsequently, these details are "injected" into the MSI to form a pansharpened image [4, Section III].

Unfortunately, the measurement of the the response function of the multispectral sensor that is made before launch may differ from the real response function after it, which poses some problems [4, Section I.A]. We propose a method to estimate a low-pass filter that does not require the use of before-launch measurements. In fact, we estimate the relative spatial response between the multispectral and panchromatic sensors directly from the observed data, and used this estimate as the low-pass filter used to extract the spatial details of the PAN.

Let $\widetilde{\mathbf{Y}}_m$ be a version of the MSI upsampled to the spatial resolution of the PAN and let $\mathbf{p}_e \in \mathbb{R}^{n_p}$ denote an approximation of the observed PAN by linearly combining the different bands of the spatially upsampled version of the MSI. This approximation is termed the *equivalent panchromatic image* and is defined as

$$\mathbf{p}_e \triangleq \mathbf{r}^T \overline{\mathbf{Y}}_m, \tag{4.18}$$

[3] More information is available at `http://eo1.gsfc.nasa.gov/`, `http://eo1.usgs.gov/sensors/ali` and `http://eo1.usgs.gov/sensors/hyperioncoverage`.



(a) Observed panchromatic image.

(b) Observed hyperspectral image.

(c) HySure's result.

(d) GSA's result.

(e) GS's result.

(f) HPF's result.

(g) BT's result.

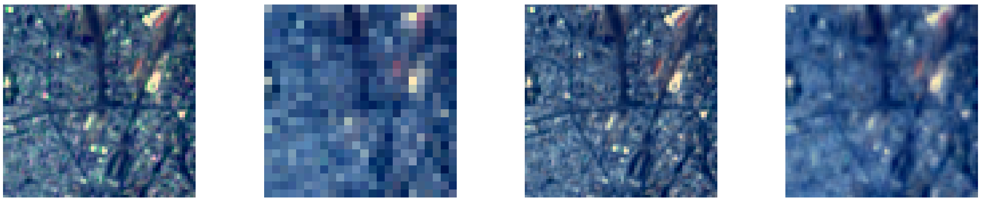
(h) FIHS's result.

(i) PCA's result.

Figure 4.2.: Results of the HSI-PAN fusion. All images, except (a), are in false color.

where $\overline{\mathbf{Y}}_m$ is obtained by stacking $\widetilde{\mathbf{Y}}_m$ with a row vector composed of all ones, i.e., $\overline{\mathbf{Y}}_m = \begin{bmatrix}\widetilde{\mathbf{Y}}_m \\ \mathbf{1}^T\end{bmatrix}$. Note that $\mathbf{r}$ is unkown and needs to be estimated. Additionally, let $\mathbf{h} \in \mathbb{R}^{n_p}$ denote the low-pass filter to be estimated. We assume this filter to have unit DC and to have finite support, which are reasonable assumptions for real-world filters.[4] The filtered version of the observed PAN is given by $\mathbf{Y}_p\mathbf{h}$, which represents in matrix-vector notation the linear convolution between the panchromatic image $\mathbf{y}_p$ and $\mathbf{h}$.

[4] This is similar to what is done in Problem (4.12).



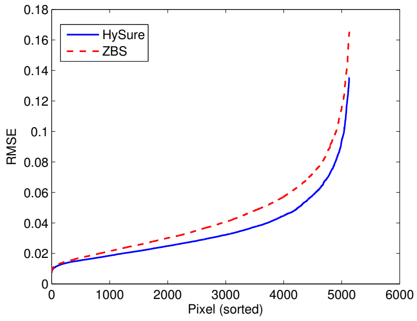
(a) Observed multispectral image.

(b) Observed hyperspectral image.

(c) HySure's result.

(d) ZBS's result.

Figure 4.3.: Results of the HSI-MSI fusion. All images are in false color. Again, Figs. 4.3c and 4.3d are very similar to Fig. 4.3a ldue to the false color rendering, but they have 128 bands, while Fig. 4.3a has only nine.



(a) RMSE between the ground truth and the results of both methods, per pixel.

Figure 4.4.: Results of the HSI-MSI fusion. The results are in ascending order.

The method used to estimate both $\mathbf{h}$ and $\mathbf{r}$ hinges on the fact that, under ideal conditions, $\mathbf{p}_e$ and $\mathbf{Y}_p\mathbf{h}$ are the same [4, Section II]. Based on this, we formulate the following problem:

$$\begin{aligned}&\underset{\mathbf{h},\mathbf{r}}{\text{minimize}} \;\; \left\|\mathbf{r}^T\widetilde{\mathbf{Y}}_m - \mathbf{Y}_p\mathbf{h}\right\|_2^2 + \lambda\left\|\mathbf{h}\right\|_2^2 + \mu\left(\left\|\mathbf{D}_v\mathbf{h}\right\|_2^2 + \left\|\mathbf{D}_h\mathbf{h}\right\|_2^2\right)\\&\text{subject to } \mathbf{h}^T\mathbf{1} = 1,\;\; \mathbf{h}\in\mathcal{H},\end{aligned} \tag{4.19}$$

where matrices $\mathbf{D}_h$ and $\mathbf{D}_v$ stand for the first-order finite differences operator in the horizontal and vertical directions, respectively, the constraint $\mathbf{h}^T\mathbf{1} = 1$ is used to impose unit DC gain, the constraint $\mathbf{h}\in\mathcal{H}$ is used to impose that $\mathbf{h}$ has a finite support—$\mathcal{H}$ is the set of filters with limited support of a given dimension—, and $\lambda > 0$ and $\mu > 0$ are two regularization parameters of the algorithm.

In order to solve this optimization problem, we follow an alternating minimization scheme. For details on the implementation of the method, see Vivone et al. [4, Algo. 1]. The proposed technique was compared with the standard one, and found to produce better results [4, Section IV].

4.4. Hyperspectral superresolution of locally low-rank images from complementary multisource data

This section introduces the contributions made by the writer in the context of the work of Veganzones et al. [8]. It takes advantage of the fact that hyperspectral images usually lie in a low-dimensional space to design a method that address the problem of fusing hyperspectral and multispectral images. Methods based on this fact perform at the level of the state of the art if the HSIs actually lie in a low-dimensional space. However, if the dimensionality of this space is not low, in the sense that it is larger than the number of multispectral bands, the performance of these methods decreases. In Veganzones et al. [8], we propose a local approach to cope with this difficulty by exploiting the fact that real-world HSIs are locally low rank since, in a small spatial neighbourhood, the number of different materials is typically also small.

As was the case of the work discussed in Section 4.2, the method that we propose was also built around the standard linear observation model for HSIs and MSIs (4.1). This work also considers the problem of spectral unmixing, which is discussed in Subsection 1.4.1.

Many algorithms that fuse HSIs and MSIs take a spectral-unmixing approach to the problem [8, Section I.A]. By this, we mean that these algorithms make two assumptions: (a) that each pixel of $\mathbf{Z}$ can be described by a mixture of a small number of "pure" spectral signatures, and (b) that, since HSIs and MSIs are obtained from the same scene, the observed materials are also the same. These algorithms then work as follows. First, since the observed HSI has a high spectral resolution, they use a spectral unmixing algorithm to extract the spectral information from the HSI [8, Section II]. Let $\mathbf{D} \in \mathbb{R}^{L_h \times n_d}$ be a matrix containing this information, i.e., containing in its columns the spectral responses of the different n_d materials. Since the number of endmembers present in a given pixel is small, we can assume that the columns of $\mathbf{Z}$ are sparsely represented as linear combinations of the columns of $\mathbf{D}$. In other words,

$$[\mathbf{Z}]_i = \mathbf{D}\boldsymbol{\alpha}_i, \tag{4.20}$$

where $[\mathbf{Z}]_i$ is a column of $\mathbf{Z}$ and $\boldsymbol{\alpha}_i \in \mathbb{R}^{n_d}$ is sparse vector that serves a similar function as the vector that contains the abundance fractions of the endmembers in Eq. (1.17). By using Eq. (4.20) and Eqs. (4.1), we can write

$$\mathbf{Y}_m = \mathbf{RDA} + \mathbf{N}_m, \tag{4.21}$$

where $\mathbf{A} \triangleq [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ is an abundance fraction matrix. By solving Eq. (4.21) to find $\mathbf{A}$, we can obtain an estimate of $\mathbf{Z}$ by using Eq. (4.20).

The success of these approaches depends fundamentally on the ability to find $\mathbf{A}$ by using Eq. (4.21). This may be hard to do, since the matrix $\mathbf{RD} \in \mathbb{R}^{L_m \times n_d}$ is often fat ($L_m < n_d$), yielding an undetermined system of equations.

The difficulties mentioned in the previous section may be circumvented by tackling the fusion problem locally instead of globally, i.e., by partitioning the image into patches, and by following the approach just described independently for each patch. In this way, if in each patch the number of materials is lower than the number of multispectral bands, the system is not undetermined anymore.

We explore two alternatives to partition the image into patches: the use of sliding windows and of binary partition trees [8, Sections III and IV, respectively]. By approaching the problem locally, we find local dictionaries and abundance matrices. Denote these by $\mathbf{D}_j$ and $\mathbf{A}_j$, respectively, where $j \in \{1, \ldots, J\}$ refers to each patch and J is the number of patches. We propose to estimate the matrix $\mathbf{A}_j$ via the problem

$$\min_{\mathbf{A}_j \geq 0} \|\mathbf{Y}_{m,j} - \mathbf{RD}_j\mathbf{A}_j\|_F^2, \quad j \in \{1, \ldots, J\} \tag{4.22}$$

where $\mathbf{Y}_{m,j}$ corresponds to the multispectral pixels in patch j. For this problem to be well-posed, the number of induced endmembers in the dictionary should be lower or equal to the number of multispectral bands, i.e., it should satisfy the condition $n_d \leq L_m$. Note that the matrix $\mathbf{R}$ is not assumed to be known beforehand and was estimated by adapting the technique described in Subsection 4.2.2 [8, Section V].

We showed through experiments, using synthetic and real datasets, that the proposed local approaches outperform the standard approach [8, Sections VI and VII].

4.5. Dissemination and addendum

The methods described in Section 4.2 are detailed in a journal paper [5]. Additionally, an earlier conference paper [3] was published with a preliminary version of this work and a different set of experiments. Furthermore, this method was included in a review paper by Loncan et al. [6, 7] in the context of the fusion of panchromatic and hyperspectral images. The MATLAB code used to generate the experiments in Simões et al. [5] is available at `https://github.com/alfaiate/HySure` and the code used to generate the experiments in Locan et al. [7] is available at `http://www.openremotesensing.net/index.php/codes/11-pansharpening/6-hyperspectral-pansharpening-a-review`.

The method discussed in Section 4.3 was published in a journal paper [4]. The method of Section 4.4 is detailed in a journal paper [8]; additionally, earlier conference papers [2, 1] were published with preliminary versions of this work and with a different set of experiments.

In Simões et al., [5, Eq. (23)] needs to be corrected. It should be

$$\begin{aligned}\mathbf{X}^{(k+1)} =& \Big[\Big(\mathbf{V}_1^{(k)} + \mathbf{A}_1^{(k)}\Big)\mathbf{B}^T + \Big(\mathbf{V}_2^{(k)} + \mathbf{A}_2^{(k)}\Big) \\ &+ \Big(\mathbf{V}_3^{(k)} + \mathbf{A}_3^{(k)}\Big)\mathbf{D}_h^T + \Big(\mathbf{V}_4^{(k)} + \mathbf{A}_4^{(k)}\Big)\mathbf{D}_v^T\Big] \\ &\times \Big[\mathbf{B}\mathbf{B}^T + \mathbf{I} + \mathbf{D}_h\mathbf{D}_h^T + \mathbf{D}_v\mathbf{D}_v^T\Big]^{-1}.\end{aligned} \tag{4.23}$$

After the publication of Simões et al. [5], a number of new methods addressing either the same problem [111, 112, 113, 114, 115, 116, 117, 118, 119] or variations of it [120] have been published as well. Wei et. al [97, 98] proposed a method which is able to deal with the terms involving the downsampling operator $\mathbf{M}$ by using the FFT directly.

Yokoya et al. [121] compared HySure with other fusion algorithms, including some that were published after this method [111, 97, 114] in the context of the problem of fusing multispectral and hyperspectral images. HySure was found to be the one that "showed the most consistent and high performance in all tests."

4.6. Conclusions

We discussed a method proposed by us in Simões et al. [5]. The method, termed HySure, is used to perform the fusion of hyperspectral images with either panchromatic or multispectral ones, with the goal of obtaining images that have high resolution in both the spatial and the spectral domains. This problem is closely related to the pansharpening one, but presents new challenges due to the much larger size of hyperspectral images when compared with the multispectral images normally used in pansharpening, and to the fact that the different images do not normally have a complete spectral overlap. In addition to performing the fusion, the proposed method is also able to estimate the relative spectral and spatial responses of the sensors from the data. We formulated the fusion problem as a convex program, solved via the SALSA—an instance of the ADMM. The estimation of the relative responses of the sensors was formulated as a convex quadratic program. Taking advantage of the low intrinsic dimensionality of hyperspectral images by working on a subspace of the space where those images are defined, and using an adequate variable splitting, we obtained an effective algorithm which compared quite favorably with several published methods on both simulated and real-life data.

We also discussed a method, proposed by us in Vivone et al. [4], that is used in the pansharpening algorithms based on the detail-injection model. In that work, we estimated the relative spectral response between the multispectral and panchromatic sensors directly from the observed data, and used this estimate as the low-pass filter used to extract the spatial details of the PAN. We provided some experimental evidence of its performance, and the method compared favorably to the state of the art.

Finally, we discussed our work in Veganzones et al. [8], where two methods for hyperspectral super-resolution via local-dictionary learning using spectral-unmixing algorithms were proposed. The experimental results showed that the proposed approaches are useful for the estimation of superresolution HSIs from locally low-rank HSIs and MSIs, even if the actual spectral response is unknown and needs to be estimated from the data.

5. Optimization algorithms

This chapter presents two results in the field of optimization that are yet to be published. The first contribution is related to the concept of operator-weighted averaged operators, and to the study of the asymptotic behavior of fixed-point iterations of these operators. The second contribution is a study of the connections between ADMM and a second-order primal–dual method when solving ℓ^2+regularizer minimization problems.

Contents

5.1. Introduction

This chapter is divided into two parts. In the first one, we introduce a certain type of operator, and establish some connections with it and a number of algorithms, in particular the ones used to solve minimization problems with sparsity-inducing regularizers. Many of these methods are generic in the sense that they do not take into account the sparsity of the solution in any particular way. However, some of them, such as semismooth Newton methods, are able to take advantage of this sparsity to accelerate their convergence. In this chapter, we show how to extend these algorithms in different directions, and we study the convergence of the resulting algorithms in real Hilbert spaces. We base our analysis on a variation on the well-known Krasnosel'skiĭ–Mann scheme and show that these methods are a particular case of this variation. Some alternative interpretations of these methods, as well as some applications, are also discussed. In particular, the methods are experimentally shown to be able to achieve substantially faster convergence rates than standard first- and second-order methods when solving a simple problem.

In the second part, we establish some connections between ADMM and a second-order primal–dual method. ADMM has been widely used in recent years to solve large-scale problems in signal processing and machine learning applications. This is due to its ability to find solutions to convex minimization problems in a computationally efficient way in many situations of practical interest. ADMM is a first-order method, and finds solutions at a linear convergence rate under certain assumptions. However, when using ADMM to solve ℓ^2+regularizer minimization problems, one is able to achieve faster rates, in practice, than when using first-order methods such as the forward–backward one. In the literature, this was suggested to be a consequence of implicitly using second-order information about the function being minimized, in a way akin to the use of this type of information in second-order methods such as proximal Newton ones. In this chapter, we substantiate this suggestion by showing that ADMM shares some similarities with a particular instance of a variable-metric primal–dual method, under certain conditions, when used to solve ℓ^2+regularizer minimization problems.

In the interest of readability, we defer the proofs of all the propositions, theorems, and corollaries of this chapter to Section 5.4.

5.1.1. Outline and contributions

The structure of this chapter is as follows. Section 5.2 corresponds to the first contribution (operator-weighted averaged operators). We define operator-weighted averaged operators, and show that they have a contractive property. In Subsection 5.2.1, we study the asymptotic behavior of fixed-point iterations of these operators, and prove the convergence of these iterations under certain conditions. We base ourselves on the fact that they produce a sequence that is Fejér monotone (specifically, variable-metric Fejér monotone). In Subsection 5.2.2, we show how operator-weighted averaged operators can be used to extend an existing algorithm from the literature—a variable-metric forward–backward method, which itself encompasses many other algorithms—, and we prove its convergence under certain conditions. In Subsection 5.2.3, we show how this extension can be used to solve a primal–dual problem first studied by Combettes and Pesquet [122], which generalizes many problems (most notably convex ones). In Subsection 5.2.4, we discuss a simple application of the proposed method to solve an inverse problem, and discuss a framework that incorporates existing active-set methods in order to solve ℓ_2-regularized minimization problems. We prove its convergence under certain conditions by showing that this framework is a particular instance of an algorithm discussed in a previous subsection. Section 5.3 corresponds to the second contribution (ADMM as a second-order method). Subsection 5.3.1 discuss a variable-metric primal–dual method. Subsection 5.3.2 details how this method and ADMM can be used to solve ℓ^2+regularizer minimization problems, and compares and contrasts both methods. Subsection 5.3.3 provides some experimental results comparing the two methods. Section 5.4 provides the proofs of all the propositions, theorems, and corollaries of this chapter. Section 5.5

concludes.

Although the idea behind operator-weighted averaged operators is quite natural, it has not, to the best of our knowledge, been formalized before. The different interpretations given in this chapter seem to be new. The asymptotic analysis of the fixed-point iterations of these operators seems to be new; it is based on some previous results on the convergence of variable-metric quasi-Fejér sequences [123, 81]. The framework that incorporates existing active-set methods also seems to be new. The variable-metric primal–dual method introduced in Subsection 5.3.1 appears to be new, as well as the comparison between it and ADMM when used to solve ℓ^2+regularizer minimization problems.

5.2. Operator-weighted averaged operators

As discussed in Section 2.5, it is possible to analyze many algorithms used to solve convex optimization problems through the study of monotone-operator theory. In fact, all the algorithms discussed in Chapter 2, except for semismooth Newton methods, can be seen as instances of the following scheme:

$$x^{k+1} = T_{\lambda^k} x^k \triangleq x^k + \lambda^k \left(R x^k - x^k \right), \tag{5.1}$$

which is known as the Krasnosel'skiĭ–Mann method (cf. Algorithm 11, page 36), and where R is a nonexpansive operator. For example, one can recover the forward–backward (cf. Algorithm 12, page 36) and the Douglas–Rachford methods (cf. Algorithm 13, page 37) by making $R = J_{\tau A} \circ (\mathrm{Id} - \tau C)$ and $R = (2J_{\tau A} - \mathrm{Id}) \circ (2J_{\tau B} - \mathrm{Id})$, respectively, where A and B are maximally monotone operators and C is a cocoercive operator. Note that, for every k, the operators T_{λ^k} are known as averaged operators [cf. Eq.(2.42)] if $\lambda^k \in]0, 1[$.

In this section, we propose and discuss an alternative scheme to Eq. (5.1),

$$x^{k+1} = T_{\Lambda^k} x^k \triangleq x^k + \Lambda^k \left(R x^k - x^k \right) \tag{5.2}$$

where, for every k, Λ^k is an operator in $\mathcal{X}$ such that $\mathrm{Id} \succeq \Lambda^k \succ 0$. We call the operators T_{Λ^k}, *operator-weighted averaged operators.* It is clear that, if $\Lambda = \lambda \mathrm{Id}$, we recover the original scheme [cf. Eq. (5.1)]. As before, we can consider that $R = J_{\tau A} \circ (\mathrm{Id} - \tau C)$ or that $R = (2J_{\tau A} - \mathrm{Id}) \circ (2J_{\tau B} - \mathrm{Id})$ to build extensions of the forward–backward and the Douglas–Rachford methods, respectively.

Eq. (5.2) can by interpreted in different ways. For example, it can be seen as a left-preconditioning scheme. Note that, under the convergence conditions discussed in Section 2.3, if we iterate Eq. (5.1), we can solve the problem of finding a fixed point of R, i.e.,

$$\text{find } x \in \mathcal{X} \quad \text{such that } Rx = x. \tag{5.3}$$

We can also consider problems equivalent to this one, in the sense that they share the same set of solutions, but that may be more convenient to solve. For example, we can consider a preconditioned version of this problem, i.e.,

$$\text{find } x \in \mathcal{X} \quad \text{such that } \Lambda R x = \Lambda x, \tag{5.4}$$

where Λ is an operator in $\mathcal{X}$ such that $\Lambda \succ 0$. Since, by iterating Eq. (5.1), we can solve Problem (5.3), a similar reasoning suggests that, by iterating Eq. (5.2), we can solve Problem (5.4). These iterations are very similar to some of the ones used to solve linear systems. Consider the system $\mathbf{Ax} = \mathbf{b}$. This system can be solved for $\mathbf{x}$, under certain conditions, by iterating $\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda^k \left(\mathbf{A}\mathbf{x}^k - \mathbf{b}\right)$. This iteration is known in the numerical-analysis community as the Richardson iteration, and is a particular instance of a first-order linear nonstationary iterative method (examples of others are the Jacobi and the Gauss–Seidel iterations) [124, 125]. Sometimes, for computational reasons (e.g., if $\mathbf{A}$ is deemed to be too ill conditioned), it is convenient to consider a preconditioned version of the linear system: $\mathbf{PAx} = \mathbf{Pb}$, where $\mathbf{P}$ is an invertible matrix. The corresponding preconditioned version of the Richardson iteration is $\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda^k \mathbf{P} \left(\mathbf{A}\mathbf{x}^k - \mathbf{b}\right)$. Note that the preconditioner Λ is different from the preconditioner U discussed in Section 2.5, and both can be used simultaneously. In Section 5.2.2, we consider a version of the forward–backward method that makes use of the two preconditioners. The core iteration of that version of the method is $x^{k+1} = x^k + \Lambda^k \left(J_{\tau U A} x^k \circ (x^k - \tau U C x^k) - x^k\right)$.

Eq. (5.1) can also be seen as the iteration of a line-search method if one considers $\lambda^k \left(R x^k - x^k\right)$ to be a step in the direction of the fixed-point residual $\left(R x^k - x^k\right)$ with step-length parameter λ^k.[1] The term $\Lambda^k \left(R x^k - x^k\right)$ in Eq. (5.2) can also be seen as indicating a search direction by noting the similarities with second-order line-search methods. Furthermore, by incorporating second-order information about $\left(R x^k - x^k\right)$ in Λ^k, we recover a Newton-like method. In fact, as we discuss next, the semismooth Newton methods discussed in Section 2.4 can be seen as a particular case of Eq. (5.2), for a given choice of the operators Λ^k and R.

We can establish parallels between the proposed scheme and some algorithms. One of these are semismooth Newton methods. Consider Eq. (2.35). This equation can be seen as an instance of Eq. (5.2) if one makes $\Lambda^k = [V(\mathbf{x}^k)]^{-1}$ and $R = J_{\tau\partial g} \circ (\mathrm{Id} - \tau \nabla f)$, where $V(\mathbf{x}^k)$ is defined as in Eq. (2.36), $f = \|\mathbf{y} - \mathbf{H} \cdot \|^2$, and $g = \mu \| \cdot \|_1$. Note that the nonexpansive operator R given here takes the same form as the operator R in the forward–backward method (cf. Algorithms 4 and 12, pages 27 and 36, respectively). This operator is slantly differentiable (see Section 2.4) and the operator Λ^k satisfies the conditions enunciated in Line 1 of Algorithm 10. For this choice of operators, Eq. (5.2) can be seen as a Newton-like iteration. Other active-set methods [126, 127, 128, 129] can also be considered as particular examples of the proposed scheme. In those cases, the operator Λ^k takes a form different from the one of Eq. (2.36). One could also consider generalizations of second-order differentiability other than slant differentiability. An example of such a generalization is epi-differentiabililty [130, Chapter 13]. This notion was used in the work of Stella et al. [131] to design quasi-Newton versions of the forward–backward method. This generalization corresponds to yet another form of the operator Λ^k. In this section, we consider that $\mathrm{Id} \succ \Lambda^k$, for every k. The semismooth Newton methods discussed in Section 2.4 do not impose such a restriction.

We can also establish parallels between the proposed scheme and coordinate-descent

[1] Compare with Eq. (2.5). This idea has been explored by Giselsson et al. [54], where the authors considered steps with length parameters $\lambda^k \geq 1$.

methods. Consider the case where, for every k, Λ^k is a diagonal operator whose entries are either 0 or 1. These operators can then be used to select a particular coordinate—or block of coordinates—of x. Consider now that $\{\Lambda^k\}$ is a sequence of such operators and that the choice of which coordinates are 0 or 1 for every Λ^k obeys a given strategy. By iterating Eq. (5.2), we recover a method that uses the same strategy used in the so-called coordinate-descent methods. For example, if $R = \mathrm{Id} - \tau\nabla f$, we recover the coordinate-gradient-descent method.[2] One can follow different selection strategies by an appropriate choice of operator Λ^k. Since the operator Λ^k is binary, we have $\mathrm{Id} \succeq \Lambda^k \succeq 0$ and not $\mathrm{Id} \succ \Lambda^k \succ 0$, as we assumed initially. Algorithms resulting from this choice of Λ^k have been studied elsewhere (see, e.g., Combettes and Pesquet [134]) and are not analyzed in the remainder of this section.

5.2.1. An extension of the Krasnosel'skiĭ–Mann method

In this subsection, we define operator-weighted averaged operators, and prove that they have a contractive property. We then study the asymptotic behavior of fixed-point iterations of these operators. These iterations can be seen as an extension of the Krasnosel'skiĭ–Mann method.

The following definition extends the notion of averaged operators, which is given by Eq. (2.42).

Definition 5.2.1 (Operator-weighted averaged operators)**.** Let D be a nonempty subset of $\mathcal{X}$, let $\epsilon \in \,]0,1[$, and let Λ be an operator in $\mathcal{X}$ such that

$$\mu\mathrm{Id} \succeq \Lambda \succeq \alpha\mathrm{Id}, \quad \text{where } \mu,\, \alpha \in [\epsilon, 1-\epsilon]. \tag{5.5}$$

We say that an operator $T_\Lambda : D \to \mathcal{X}$ is an operator-weighted averaged operator if there exists a nonexpansive operator $R : D \to \mathcal{X}$ such that

$$T_\Lambda \triangleq (\mathrm{Id} - \Lambda) + \Lambda R. \tag{5.6}$$

The following proposition establishes a contractive property that is similar to the one of averaged operators [cf. Eq. (2.43)].

Proposition 5.2.2. *Let D be a nonempty subset of $\mathcal{X}$, let $\epsilon \in \,]0,1[$, let Λ be an operator in $\mathcal{X}$ satisfying* (5.5)*, let $R : D \to \mathcal{X}$ be a nonexpansive operator, and let $T_\Lambda : D \to \mathcal{X}$*

[2] The iterations of the coordinate-gradient-descent method are very similar to the ones of the gradient-descent method (cf. Algorithm 1, page 21). The difference is that the update in $\mathbf{x}^{k+1}$ is made for one coordinate while the others are kept fixed. For every k, the update is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \tau[\nabla f(\mathbf{x}^k)]_{i^k}\mathbf{e}_{i^k},$$

where i^k is a given coordinate and $\mathbf{e}_i$ is a binary vector whose components are zero, except for the i-th one.

For more details on coordinate- and block-coordinate-descent methods, see, e.g., Wright [132] and Shi et al. [133].

be an operator as defined in Eq. (5.6)*. Then the operator* T_Λ *is* μ*-averaged in the metric induced by* Λ^{-1}*. In other words,* T_Λ *verifies*

$$\begin{aligned} \|T_\Lambda x - T_\Lambda y\|^2_{\Lambda^{-1}} &\leq \|x - y\|^2_{\Lambda^{-1}} \\ &\qquad - \frac{1-\mu}{\mu} \|(\mathrm{Id} - T_\Lambda)\, x - (\mathrm{Id} - T_\Lambda)\, y\|^2_{\Lambda^{-1}}, \quad \forall x \in D, \quad \forall y \in D. \end{aligned}$$

Proof. We defer all proofs to Section 5.4. □

Remark 5.2.3. By making $\Lambda = \lambda\mathrm{Id}$ with $\lambda =]0,1[$, we recover the standard definition of averaged operators. In other words, $T_{\lambda\mathrm{Id}}$ is λ-averaged. Naturally, we also recover a contractive property of averaged operators [cf. Eq. (2.43)] (note that, in this case, $\alpha = \mu = \lambda$).

Consider the following algorithm, which is an extension of the Krasnosel'skiĭ–Mann method (cf. Algorithm 11, page 36).

Algorithm 17: Fixed-point iterations of T_{Λ^k}.

```
Choose $x^0 \in \mathcal{X}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\mathrm{Id} \succ \Lambda^k \succ 0$;
    $x^{k+1} \leftarrow T_{\Lambda^k} x^k = x^k + \Lambda^k \left(Rx^k - x^k\right)$;
    $k \leftarrow k + 1$;
end
```

The following theorem establishes some convergence properties of this algorithm.

Theorem 5.2.4. *Let* D *be a nonempty closed convex subset of* $\mathcal{X}$*, let* $\epsilon \in]0,1[$*, let* $\{\eta^k\} \in \ell^1_+(\mathbb{N})$*, let* $\{\Lambda^k\}$ *be a sequence of operators in* $\mathcal{X}$ *such that, for all* $k \in \mathbb{N}$*,*

$$\begin{cases} \mu^k \mathrm{Id} \succeq \Lambda^k \succeq \alpha^k \mathrm{Id}, \\ \mu^k,\, \alpha^k \in [\epsilon, 1-\epsilon] \\ \left(1 + \eta^k\right) \Lambda^{k+1} \succeq \Lambda^k, \end{cases} \tag{5.7}$$

and let $R : D \to D$ *be a nonexpansive operator such that* Fix $R \neq \emptyset$*.*

Let $x^0 \in D$ *and let* $\{x^k\}$ *be a sequence generated by Algorithm 17. Then the following hold:*

1) *Let* $x^* \in D$*. Then,* $x^* \in$ Fix R *if and only if* $x^* \in$ Fix T_{Λ^k}*.*

2) $\{x^k\}$ *is Fejér monotone with respect to* Fix R *relative to* $\left\{\left(\Lambda^k\right)^{-1}\right\}$*.*[3]

3) *The sequence* $\{Rx^k - x^k\}$ *converges strongly to* 0*.*

4) $\{x^k\}$ *converges weakly to a point in* Fix R*.*

Proof. We defer all proofs to Section 5.4. □

[3] See Footnote 6 of Chapter 2 for a definition.

5.2.2. An extension of a variable-metric forward–backward method

In this subsection, we show how operator-weighted averaged operators can be used to extend a variable-metric forward–backward method proposed by Combettes and Vũ [81].

Consider Algorithm 18 (which is an extension of Algorithm 12) to solve Problem (2.47). In what follows, $\left\{U^k\right\}, \left\{\Lambda^k\right\}$ are sequences of bounded linear operators, and $\left\{a^k\right\}, \left\{b^k\right\}$ are absolutely summable sequences that can be used to model errors.

Algorithm 18: An extension of the variable-metric forward–backward algorithm.

```
Choose $x^0 \in \mathcal{X}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\gamma^k > 0,\ U^k \succ 0,$ and $\mathrm{Id} \succ \Lambda^k \succ 0$;
    $y^k \leftarrow x^k - \gamma^k U^k \left(C x^k + b^k\right)$;
    $x^{k+1} \leftarrow x^k + \Lambda^k \left(J_{\gamma^k U^k A} y^k + a^k - x^k\right)$;
    $k \leftarrow k + 1$;
end
```

The following theorem establishes some convergence properties of this algorithm.

Theorem 5.2.5. *Let $A : \mathcal{X} \to 2^{\mathcal{X}}$ be a maximally monotone operator, let $\beta \in \left]0, +\infty\right[$, and let C be a β-cocoercive operator.*

Let $\left\{U^k\right\}$ be a sequence of operators in $\mathcal{X}$ such that, for all $k \in \mathbb{N}$,

$$\begin{cases} \mu_U \mathrm{Id} \succeq U^k \succeq \alpha_U \mathrm{Id}, \\ \mu_U,\ \alpha_U \in \left]0, +\infty\right[, \end{cases} \tag{5.8}$$

let $\epsilon \in \left]0, \min\{1, 2\beta/(\mu_U + 1)\}\right[$, let $\left\{\Lambda^k\right\}$ be a sequence of operators in $\mathcal{X}$ such that, for all k,

$$\begin{cases} \Lambda^k U^k = U^k \Lambda^k, \\ \mu \mathrm{Id} \succeq \Lambda^k \succeq \alpha \mathrm{Id}, \\ \mu,\ \alpha \in [\epsilon, 1], \end{cases} \tag{5.9}$$

let $\left\{\eta^k\right\} \in \ell^1_+(\mathbb{N})$, and let

$$(1 + \eta^k)\Lambda^{k+1} U^{k+1} \succeq \Lambda^k U^k. \tag{5.10}$$

Let $\left\{\gamma^k\right\}$ be a sequence in $[\epsilon, (2\beta - \epsilon)/\mu_U]$ and let $\left\{a^k\right\}, \left\{b^k\right\} \in \ell^1(\mathbb{N})$.

Suppose that $Z = \mathrm{zer}\ (A + C) \neq \emptyset$. Let $\left\{x^k\right\}$ be a sequence generated by Algorithm 18. Then the following hold:

1) *$\left\{x^k\right\}$ is quasi-Fejér monotone[4] with respect to Z relative to $\left\{\left(\Lambda^k U^k\right)^{-1}\right\}$.*

2) *$\left\{x^k\right\}$ converges weakly to a point in Z.*

[4] See Footnote 6 of Chapter 2 for a definition.

Proof. We defer all proofs to Section 5.4. □

Remark 5.2.6. (Variable-metric forward–backward method) By making $\Lambda^k = \lambda^k \mathrm{Id}, \forall k$, where $\{\lambda^k\}$ is a sequence in $[\epsilon, 1]$, we recover the algorithm analyzed in [81, Theorem 4.1].

Remark 5.2.7. The assumption that, for every k, U^k and Λ^k commute, i.e., $\Lambda^k U^k - U^k \Lambda^k = 0$, may seem to be severe. However, take into account that existing algorithms consider one of these operators to be the identity operator: in semismooth Newton methods, $U^k = \mathrm{Id}$, $\forall k$ and Λ^k contains some second-order information about R; in variable-metric forward–backward methods, $\Lambda^k = \mathrm{Id}$, $\forall k$. In a sense, the algorithm proposed here unifies these two approaches and allows a more flexible choice of operators.

In what follows, we give an example of the application of Algorithm 18 to solve convex problems of the form of Problem (2.15). Consider the following algorithm

Algorithm 19: An application of Algorithm 18 to solve convex problems.

```
Choose $x^0 \in \mathcal{X}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\gamma^k > 0$, $U^k \succ 0$, and $\mathrm{Id} \succ \Lambda^k \succ 0$;
    $y^k \leftarrow x^k - \gamma^k U^k \left(\nabla g(x^k) + b^k\right)$;
    $x^{k+1} \leftarrow x^k + \Lambda^k \left(\mathrm{prox}_{\gamma^k f}^{(U^k)^{-1}} y^k + a^k - x^k\right)$;
    $k \leftarrow k + 1$;
end
```

The following corollary establishes some convergence properties of Algorithm 19.

Corollary 5.2.8. *Let $f \in \Gamma_0(\mathcal{X})$, let $\beta \in]0, +\infty[$, and let $g : \mathcal{X} \to \mathbb{R}$ be convex and differentiable with a $1/\beta$-Lipschitzian gradient. Let $\epsilon \in]0, \min\{1, 2\beta/(\mu_U + 1)\}[$, let $\{\eta^k\} \in \ell^1_+(\mathbb{N})$, and let $\{U^k\}$ and $\{\Lambda^k\}$ be sequences of operators in $\mathcal{X}$ satisfying* (5.8), (5.9) *and* (5.10) *for all $k \in \mathbb{N}$. Let $\{\gamma^k\}$ be a sequence in $[\epsilon, (2\beta - \epsilon)/\mu_U]$, and let $\{a^k\}, \{b^k\} \in \ell^1(\mathbb{N})$.*

Suppose that $Z = \mathrm{Argmin}\,(f + g) \neq \emptyset$. Let $\{x^k\}$ be a sequence generated by Algorithm 19. Then the following hold:

1) *$\{x^k\}$ is quasi-Fejér monotone with respect to Z relative to $\left\{\left(\Lambda^k U^k\right)^{-1}\right\}$.*

2) *$\{x^k\}$ converges weakly to a point in Z.*

Proof. We defer all proofs to Section 5.4. □

Remark 5.2.9. When g is a sparsity-inducing regularizer and the operator Λ^k satisfies the conditions enunciated in Line 1 of Algorithm 10, for every k, we recover the semismooth Newton methods discussed in Section 2.4.

5.2.3. Primal-dual composite monotone inclusions

In this subsection, we present an algorithm that solves a primal–dual problem, which encapsulates many other problem formulations [122, 68, 81]. This problem generalizes the ones discussed in Section 2.5, namely the inclusions $0 \in Ax+Bx$ and $0 \in Ax+Cx$, where A and B are maximally monotone operators, and C is a cocoercive operator, and also generalizes others (see Vũ [68] for more examples). Since this problem is quite general, by devising an algorithm to solve it, we effectively tackle a large number of problems simultaneously. The problem involves the parallel sum of two set-valued operators, which is an operation that can be seen as a regularization of one operator by the other. When the operators involved are subdifferentials of proper lower semi-continuous convex functions, it corresponds to the infimal convolution.[5]

Let $A : \mathcal{X} \to 2^{\mathcal{X}}$ be a maximally monotone operator, let $\mu \in]0, +\infty[$, let $C : \mathcal{X} \to \mathcal{X}$ be μ-cocoercive, and let $z \in \mathcal{X}$. Let N be a strictly positive integer; for every $j \in \{1, \dots, N\}$, let $r_j \in \mathcal{V}_j$, let $B_j : \mathcal{V}_j \to 2^{\mathcal{V}_j}$ be maximally monotone, let $\nu_j \in]0, +\infty[$, let $E_j : \mathcal{V}_j \to 2^{\mathcal{V}_j}$ be maximally monotone and ν_j-strongly monotone, let $L_j \in \mathcal{B}(\mathcal{X}, \mathcal{V}_j)$ such that $L_j \neq 0$, and let ω_j be real numbers in $]0, 1]$ such that $\sum_{j=1}^{N} \omega_j = 1$. The primal–dual problem is as follows: to solve the primal inclusion

$$\text{find} \quad x \in \mathcal{X} \quad \text{such that} \quad z \in Ax + \sum_{j=1}^{N} \omega_j L_j^* \left(\left(B_j \,\square\, E_j \right) \left(L_j x - r_j \right) \right) + Cx, \tag{5.11}$$

together with the dual inclusion

$$\text{find } d_1 \in \mathcal{V}_1, \dots, d_N \in \mathcal{V}_N \text{ such that } \exists\, x \in \mathcal{X} \text{ and } \begin{cases} z - \sum_{j=1}^{N} \omega_j L_j^* d_j \in Ax + Cx, \\ d_j \in \left(B_j \,\square\, E_j \right) \left(L_j x - r_j \right), \ \forall_j \in \{1, \dots, N\}. \end{cases} \tag{5.12}$$

The sets of solutions to (5.11) and (5.12) are denoted by P and D, respectively.

Consider Algorithm 20, which is an extension of Algorithm 18, to solve this primal–dual problem. In what follows, for all j, $\left\{U^k\right\}, \left\{\Lambda^k\right\}, \left\{U_j^k\right\}, \left\{\Lambda_j^k\right\}$ are sequences of bounded linear operators, and $\left\{a^k\right\}, \left\{b_j^k\right\}, \left\{c^k\right\}, \left\{e_j^k\right\}$ are absolutely summable sequences that can be used to model errors.

[5] See Footnote 3 of Chapter 2 for a brief discussion on the connections between the infimal convolution and regularization.

Algorithm 20: An extension of the variable-metric primal–dual Algorithm.

```
Choose $x^0 \in \mathcal{X}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    for $j \leftarrow 1, \dots, N$ do
        Choose $U_j^k \succ 0$ and $\mathrm{Id} \succ \Lambda_j^k \succ 0$;
        $q_j^k \leftarrow J_{U_j^k B_j^{-1}} \left( d_j^k + U_j^k \left( L_j x^k - E_j^{-1} d_j^k - e_j^k - r_j \right) \right) + b_j^k$;
        $y_j^k \leftarrow 2q_j^k - d_j^k$;
        $d_j^{k+1} \leftarrow d_j^k + \Lambda_j^k \left( q_j^k - d_j^k \right)$;
    end
    Choose $U^k \succ 0$, and $\mathrm{Id} \succ \Lambda^k \succ 0$;
    $p^k \leftarrow J_{U^k A} \left( x^k - U^k \left( \sum_{j=1}^N \omega_j L_j^* y_j^k + Cx^k + c^k - z \right) \right) + a^k$;
    $x^{k+1} \leftarrow x^k + \Lambda^k \left( p^k - x^k \right)$;
    $k \leftarrow k + 1$;
end
```

The following corollary establishes some convergence properties of Algorithm 20.

Corollary 5.2.10. *Suppose that*

$$z \in \operatorname{ran} \left(A + \sum_{j=1}^{N} \omega_j L_j^* \left(\left(B_j \,\Box\, E_j \right) \left(L_j \cdot - r_j \right) \right) + C \right) \tag{5.13}$$

and set

$$\beta \triangleq \min\{\mu, \nu_1, \dots, \nu_N\}. \tag{5.14}$$

Let $\left\{U^k\right\}$ be a sequence of operators in $\mathcal{X}$ such that, for all $k \in \mathbb{N}$,

$$\begin{cases} \mu_U \mathrm{Id} \succeq U^k \succeq \alpha_U \mathrm{Id}, \\ \mu_U,\ \alpha_U \in \left]0, +\infty\right[, \end{cases} \tag{5.15}$$

let $\epsilon \in \left]0, \min\{1, \beta\}\right[$, let $\left\{\Lambda^k\right\}$ be a sequence of operators in $\mathcal{X}$ such that, for all k,

$$\begin{cases} \Lambda^k U^k = U^k \Lambda^k, \\ \mu \mathrm{Id} \succeq \Lambda^k \succeq \alpha \mathrm{Id}, \\ \mu,\ \alpha \in [\epsilon, 1], \end{cases} \tag{5.16}$$

and let

$$\Lambda^{k+1} U^{k+1} \succeq \Lambda^k U^k. \tag{5.17}$$

Additionally, for every $j \in \{1, \dots, N\}$, let $\left\{U_j^k\right\}$ be a sequence of operators in $\mathcal{V}_j$ such that, for all $k \in \mathbb{N}$,

$$\begin{cases} \mu_U \mathrm{Id} \succeq U_j^k \succeq \alpha_U \mathrm{Id}, \\ \mu_U,\ \alpha_U \in \left]0, +\infty\right[, \end{cases} \tag{5.18}$$

let $\left\{\Lambda_j^k\right\}$ *be a sequence of operators in* $\mathcal{V}_j$ *such that, for all* k*,*

$$\begin{cases}\Lambda_j^k U_j^k = U_j^k \Lambda_j^k, \\ \mu \mathrm{Id} \succeq \Lambda_j^k \succeq \alpha \mathrm{Id}, \\ \mu,\, \alpha \in [\epsilon, 1], \end{cases} \tag{5.19}$$

and let

$$\Lambda_j^{k+1} U_j^{k+1} \succeq \Lambda_j^k U_j^k. \tag{5.20}$$

Let, for all j*,* $\left\{a^k\right\}, \left\{b^k\right\}, \left\{c_j^k\right\}, \left\{e_j^k\right\} \in \ell^1(\mathbb{N})$*.*

For every k*, set*

$$\delta^k \triangleq \left(\sum_{j=1}^N \omega_j \left\|\sqrt{U_j^k} L_j \sqrt{U^k}\right\|^2\right)^{-\frac{1}{2}} - 1 \tag{5.21}$$

and suppose that

$$\xi^k \triangleq \frac{\delta^k}{(1+\delta^k)\mu_U} \geq \frac{1}{2\beta - \epsilon}. \tag{5.22}$$

Let $\left\{x^k\right\}$ *be a sequence generated by Algorithm 20. Then the following hold:*

1) x^k *converges weakly to a point in* P*.*

2) $\left(d_1^k, \ldots, d_N^k\right)$ *converges weakly to a point in* D*.*

Proof. We defer all proofs to Section 5.4. □

Convex minimization problems

The previously described primal–dual problem formulation can be used to formulate convex primal–dual minimization problems [122, 68, 81]. Let $g \in \Gamma_0(\mathcal{X})$, let $\mu \in]0, +\infty[$, let $f : \mathcal{X} \to \mathbb{R}$ be convex and differentiable with a μ^{-1}-Lipschitzian gradient, and let $z \in \mathcal{X}$. Let N be a strictly positive integer; for every $j \in \{1, \ldots, N\}$, let $r_j \in \mathcal{V}_j$, let $h_j \in \Gamma_0(\mathcal{V}_j)$, let $\nu_j \in]0, +\infty[$, let $l_j \in \Gamma_0(\mathcal{V}_j)$ be ν_j-strongly convex, let $L_j \in \mathcal{B}(\mathcal{X}, \mathcal{V}_j)$ such that $L_j \neq 0$, and let ω_j be real numbers in $]0, 1]$ such that $\sum_{j=1}^N \omega_j = 1$. In Problems (5.11) and (5.12), making, for every j, $A = \partial g$, $C = \nabla \mu$, $B_j = \partial h_j$, and $E_j = \partial l_j$, yields the following primal problem [122, Theorem 4.2],

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad g(x) + \sum_{j=1}^N \omega_j \left(h_j \star_{\inf} l_j\right)\left(L_j x - r_j\right) + f(x) - \langle x, z\rangle, \tag{5.23}$$

together with its corresponding dual problem,

$$\underset{d_1 \in \mathcal{V}_1, \cdots, d_j \in \mathcal{V}_j}{\text{minimize}} \quad \left(g^* \star_{\inf} h^*\right)\left(z - \sum_{j=1}^N \omega_j L_j^* d_j\right) + \sum_{j=1}^N \omega_j \left(h_j^*(d_j) + l_j^*(d_j) + \langle d_j, r_j\rangle\right). \tag{5.24}$$

The sets of solutions to (5.23) and (5.24) are denoted by P and D, respectively. The different optimization problems discussed in Chapter 2 are special cases of this primal–dual problem [122, Example 1.6].

Consider Algorithm 21 (which is an instance of Algorithm 20) to solve this primal–dual problem. In what follows, for all j, $\{U^k\}, \{\Lambda^k\}, \left\{U_j^k\right\}, \left\{\Lambda_j^k\right\}$ are sequences of bounded linear operators, and $\{a^k\}, \left\{b_j^k\right\}, \{c^k\}, \left\{e_j^k\right\}$ are absolutely summable sequences that can be used to model errors.

Algorithm 21: An application of Algorithm 20 to solve convex problems.

```
Choose $x^0 \in \mathcal{X}$ and $d_1^0 \in \mathcal{V}_1, \cdots, d_j^0 \in \mathcal{V}_j$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    for $j = 1, \dots, N$ do
        Choose $U_j^k \succ 0$, and $\mathrm{Id} \succ \Lambda_j^k \succ 0$;
        $q_j^k = \operatorname{prox}_{h_j^*}^{(U_j^k)^{-1}} \left(d_j^k + U_j^k \left(L_j x^k - \nabla l_j^*(d^k) - e_j^k - r_j\right)\right) + b_j^k$;
        $y_j^k = 2q_j^k - d_j^k$;
        $d_j^{k+1} = d_j^k + \Lambda_j^k \left(q_j^k - d_j^k\right)$;
    end
    Choose $U^k \succ 0$, and $\mathrm{Id} \succ \Lambda^k \succ 0$;
    $p^k = \operatorname{prox}_g^{(U^k)^{-1}} \left(x^k - U^k \left(\sum_{j=1}^N \omega_j L_j^* y_j^k + \nabla f(x^k) + c^k - z\right)\right) + a^k$;
    $x^{k+1} = x^k + \Lambda^k \left(p^k - x^k\right)$;
    $k \leftarrow k + 1$;
end
```

The following corollary establishes some convergence properties of Algorithm 21.

Corollary 5.2.11. *Suppose that*

$$z \in \operatorname{ran} \left(\partial g + \sum_{j=1}^N \omega_j L_j^* \left(\partial h_j \star_{inf} \partial l_j\right)\left(L_j \cdot - r_j\right) + \nabla f\right) \tag{5.25}$$

and set $\beta = \min\{\mu, \nu_1, \dots, \nu_N\}$. *Let* $\epsilon \in \left]0, \min\{1, \beta\}\right[$ *and let, for all* j, $\{U^k\}$, $\{\Lambda^k\}$, $\left\{U_j^k\right\}$, *and* $\left\{\Lambda_j^k\right\}$ *be sequences of operators satisfying* (5.15)-(5.20) *for all* $k \in \mathbb{N}$. *Let, for all* j, $\{a^k\}, \{b^k\}, \left\{c_j^k\right\}, \left\{e_j^k\right\} \in \ell^1(\mathbb{N})$. *For every* $k \in \mathbb{N}$, *set* (5.21) *and suppose that* (5.22) *holds. Let* $\{x^k\}$ *be a sequence generated by Algorithm 21. Then the following hold:*

1) x^k *converges weakly to a point in* P.

2) $\left(d_1^k, \dots, d_N^k\right)$ *converges weakly to a point in* D.

Proof. We defer all proofs to Section 5.4. □

Remark 5.2.12. By making, for every k and for every j, $z = 0$, $r_j = 0$, $U^k = \tau\mathrm{Id}$ (where $\tau > 0$), $U_j^k = \sigma\mathrm{Id}$ (where $\sigma > 0$), $\Lambda^k = \theta^k\mathrm{Id}$ (where $\{\theta^k\}$ is a sequence of positive parameters), and

$$l_j : y \to \begin{cases} 0 & \text{if } y = 0, \\ +\infty & \text{otherwise,} \end{cases} \tag{5.26}$$

we recover [67, Algorithm 5.2].

5.2.4. Applications and experiments

In this subsection, we discuss a simple problem that can be solved via Algorithm 21, and a framework that can be used to adapt existing semismooth Newton and other active-set methods to solve ℓ_2-regularized minimization problems with more than one regularizer and/or with constraints.

A simple problem

Consider the problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \|\mathbf{b} - \mathbf{H}\mathbf{x}\|^2 + \mu\|\mathbf{x}\|_1 + \delta_{[c,d]^n}(\mathbf{x}), \tag{5.27}$$

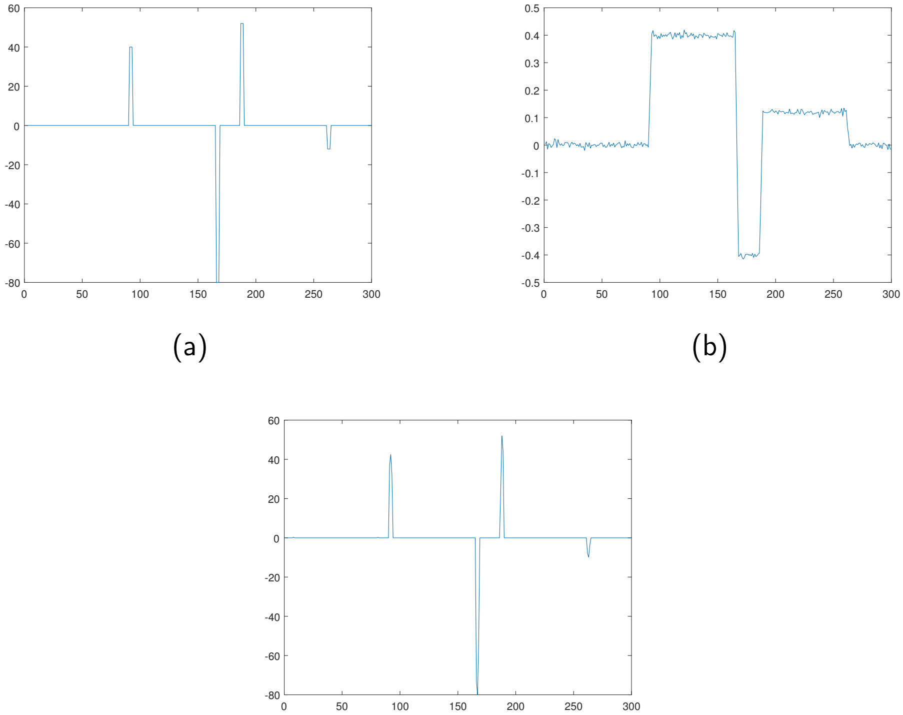
where $\mathbf{b} \in \mathbb{R}^n$, $c \in \mathbb{R}$, $d \in \mathbb{R}$, $\mu > 0$, $\delta_{[c,d]^n}(\cdot)$ is defined as in (1.14), and

$$\mathbf{H} \in \mathbb{R}^{n\times n} : \mathbf{H} = \frac{1}{N}\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & & \ddots & 0 \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{5.28}$$

Griesse and Lorenz studied this problem in the context of inverse integration [79, Section 4.1] but did not consider the term $\delta_{[c,d]^n}(\mathbf{x})$. This problem can be solved via Algorithm 21 by letting $\gamma > 0$, $\tau > 0$, by making $N = 1, \mathcal{X} = \mathbb{R}^n, \mathcal{V} = \mathbb{R}^n, L_1 = \mathbf{I}_n, r_1 = \mathbf{0}, z = \mathbf{0}$, and $\forall_k, U_1^k = \gamma\mathbf{I}_n, U^k = \tau\mathbf{I}_n, e_1^k = \mathbf{0}, b_1^k = \mathbf{0}, \Lambda_1^k = \mathbf{I}_n, c^k = \mathbf{0}, a^k = \mathbf{0}$, $f = \|\mathbf{b} - \mathbf{H}\cdot\|^2$, $g = \mu\|\cdot\|_1$, $h = \delta_{[c,d]^n}(\cdot)\, f$, if (5.26) is satisfied for $j = 1$, and by considering that Λ^k is defined similarly to Eqs. (2.35-2.38).

We recover a similar example to the one studied by Griesse and Lorenz: let the original $\mathbf{x}$ be as in Fig. 5.1a and the observed $\mathbf{b}$ be as in Fig. 5.1b, where Gaussian noise with SNR of 30 dB was added to $\mathbf{Hx}$. Set $\mu = 3 \times 10^{-3}$, $c = -80$ and $d = 52$. We compared Algorithm 21 (denoted in what follows as *Proposed*) with Algorithm 7 (denoted by ADMM) and with Algorithm 8 (denoted by CM) to solve the problem under consideration.[6] We manually tuned the different parameters of the three methods in order

[6] CM is a first-order method, since it only uses first-order information about the objective function of Problem (5.27). ADMM is, by definition, a first-order method, although, as discussed in Section 5.3, it behaves very similarly to a second-order method when used to solve ℓ_2-regularized minimization problems.



(a)

(b)

(c)

Figure 5.1.: Original **x** (a), observed **b** (b), and estimated **x** (c).

to achieve the fastest convergence results in practice. We arbitrarily chose the result of ADMM after 10^7 iterations as representative of the solution given by the three methods. Fig. 5.1c shows the results of the Proposed method and Fig. 5.2 illustrates the behavior of the three methods by showing the RMSE as a function of time, between the estimates of each method and the representative solution. The experiments were performed using MATLAB on an Intel Core i7 CPU running at 3.20 GHz, with 32 GB of RAM.

In this example, we did not enforce in any particular way the assumptions on Λ, i.e., assumptions (5.16) and (5.17), but we verified in practice that they were satisfied. However, in more complex examples, it would be necessary to devise a strategy that generates a sequence $\left\{\Lambda^k\right\}$ satisfying these assumptions. This is akin to the necessity of devising globalization strategies in other Newton-like methods [39, Chapter 7].

It is clear that the proposed method has a much faster convergence than either CM or ADMM. In practice, this improvement in convergence is similar to the one observed in the methods discussed in Section 2.4. This is due to the fact that this method requires one to compute the solution of a linear system with a much lower dimensionality than the system in ADMM. In general, the sparser the solution is, the faster the method is as well. In order to benefit from this property, we must be able to efficiently solve the lower-dimensional system, in the sense that we can pick at will which coordinates of the system are active. This may not always be possible: for example, in problems that involve

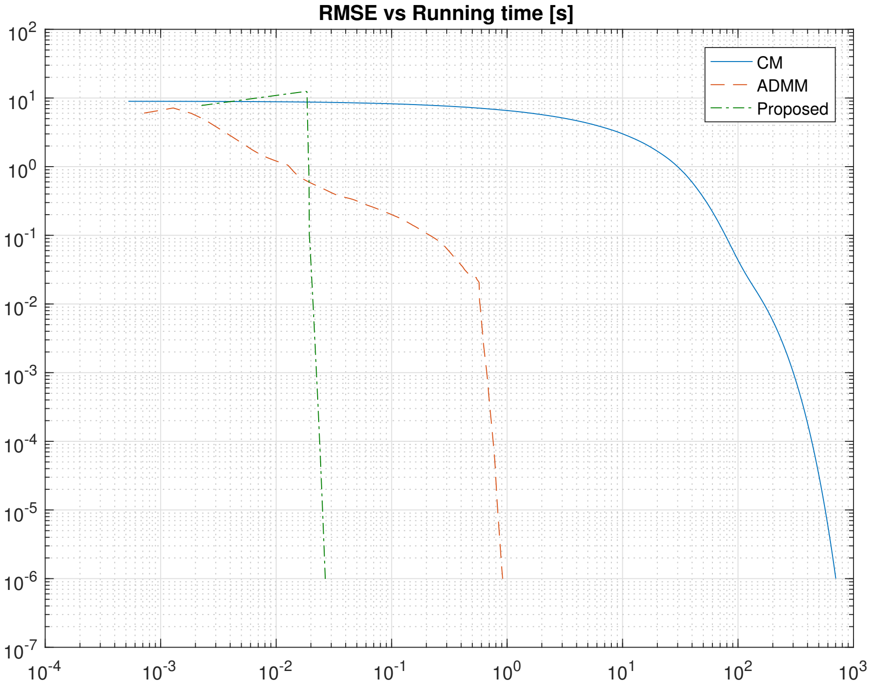


Figure 5.2.: RMSE, as a function of time, between the estimates of each iteration and the representative solution, for the three methods.

computations with the FFT of a signal, we usually have only modest improvements in speed if we wish to compute only selected elements of the FFT.[7]

Plug-and-play methods

Recently, a number of works have been published in the literature that explore the idea of substituting one of the steps of a given algorithm by another algorithm. For example, one can use the plug-and-play ADMM to solve ℓ_2-regularized minimization problems. In this method, the user substitutes one of the steps of the method by a denoising algorithm [99, 136]. We explore a similar idea by considering the use of a semismooth Newton or other active-set method to replace one of the steps of ADMM. While the goal with the use of denoising algorithms is to improve the SNR of the estimates, the goal here is improve the convergence rate.

[7] See `http://www.fftw.org/pruned.html` for details. However, for large-scale problems and for highly-sparse signals, methods known as sparse FFTs [135] may be useful (see `https://groups.csail.mit.edu/netmit/sFFT/` for details).

Consider problems of the form

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\left\|\mathbf{H}\mathbf{x}-\mathbf{b}\right\|^2 + g(\mathbf{x}) + h(\mathbf{x}), \tag{5.29}$$

where $g(\mathbf{x}) \in \Gamma_0(\mathbb{R}^n)$ is a sparsity-inducing regularizer and $h(\mathbf{x}) \in \Gamma_0(\mathbb{R}^n)$. Examples of such functions are $g(\mathbf{x}) = \mu\left\|\mathbf{x}\right\|_1$ and $h(\mathbf{x}) = \delta_{[c,d]^n}(\mathbf{x})$. Existing semismooth Newton methods do not consider the existence of a separate term h, and, in general, are not able to solve this problem efficiently, since that would require the computation of $\operatorname{prox}_{(g+h)}$. In what follows, we discuss an algorithm that can be used to solve Problem (5.29), that does not require that computation, and that allows the use of an existing semismooth Newton method in a plug-and-play fashion. Problem (5.29) can be rewritten as

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^n,\mathbf{v}\in\mathbb{R}^n}{\text{minimize}} && \frac{1}{2}\left\|\mathbf{H}\mathbf{x}-\mathbf{b}\right\|^2 + g(\mathbf{x}) + h(\mathbf{v}) \\ &\text{subject to} && \mathbf{x} = \mathbf{v}, \end{aligned} \tag{5.30}$$

which, in turn, has the augmented Lagrangian

$$\mathcal{L}(\mathbf{x},\mathbf{v},\mathbf{d}) = \frac{1}{2}\left\|\mathbf{H}\mathbf{x}-\mathbf{b}\right\|^2 + g(\mathbf{x}) + h(\mathbf{v}) + \frac{\gamma}{2}\left\|\mathbf{d}+\mathbf{x}-\mathbf{v}\right\|^2, \tag{5.31}$$

where $\mathbf{d} \in \mathbb{R}^n$ and $\gamma > 0$. Cycling through $\mathbf{x}$, $\mathbf{v}$, and $\mathbf{d}$ yields ADMM:

$$\mathbf{x}^{k+1} \in \underset{\mathbf{x}\in\mathbb{R}^n}{\arg\min}\; g(\mathbf{x}) + \frac{1}{2}\left\|\begin{bmatrix}\mathbf{H}\\ \sqrt{\gamma}\mathbf{I}_n\end{bmatrix}\mathbf{x} - \begin{bmatrix}\mathbf{b}\\ \sqrt{\gamma}(\mathbf{v}^k-\mathbf{d}^k)\end{bmatrix}\right\|^2, \tag{5.32}$$

$$\mathbf{v}^{k+1} \in \underset{\mathbf{v}\in\mathbb{R}^n}{\arg\min}\; h(\mathbf{v}) + \frac{\gamma}{2}\left\|\mathbf{d}^k+\mathbf{x}^k-\mathbf{v}\right\|^2, \tag{5.33}$$

$$\mathbf{d}^{k+1} = \mathbf{d}^k + \mathbf{x}^{k+1} - \mathbf{v}^{k+1}. \tag{5.34}$$

Problem (5.32) can be solved through one of the existing active-set methods that tackle problems with sparsity-inducing regularizers [126, 137, 138, 139, 127, 128, 129, 140]. With this in mind, and by setting $f(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{H}\mathbf{x}-\mathbf{b}\right\|^2$, we propose the following algorithm:

Algorithm 22: Plug-and-play ADMM with a semismooth Newton method.

```
Choose $\mathbf{x}^0 \in \mathbb{R}^n$, $\mathbf{v}^0 \in \mathbb{R}^n$, $\mathbf{d}^0 \in \mathbb{R}^n$, $\tau > 0$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    $\mathbf{p}^k = \operatorname{prox}_{\tau g}\left(\mathbf{x}^k - \tau\left(\nabla f(\mathbf{x}^k) + \gamma\left(\mathbf{x}^k - \mathbf{v}^k + \mathbf{d}^k\right)\right)\right)$;
    $\mathbf{x}^{k+1} = \mathbf{x}^k + \Lambda^k\left(\mathbf{p}^k - \mathbf{x}^k\right)$;
    $\mathbf{v}^{k+1} = \operatorname{prox}_{\frac{h}{\gamma}}\left(\mathbf{x}^{k+1} + \mathbf{d}^k\right)$;
    $\mathbf{d}^{k+1} = \mathbf{d}^k + \left(\mathbf{x}^{k+1} - \mathbf{v}^{k+1}\right)$;
    $k \leftarrow k + 1$;
end
```

Lines 4 and 5 of Algorithm 22 take a form similar to Algorithm 19 and can, in principle, be replaced by any of the existing active-set methods (see Remark 5.2.9). In the following corollary, we discuss some of the convergence guarantees of this algorithm by showing that it is an instance of Algorithm 21. Note that the use of any active-set method may violate the convergence conditions given here.

Corollary 5.2.13. *Suppose that*

$$\mathbf{0} \in \operatorname{ran}\ (\partial g + \partial h + \nabla f)\,. \tag{5.35}$$

Set $\beta = \|\mathbf{H}^*\mathbf{H}\|$ *and let* $\epsilon \in \,]0, \min\{1, \beta\}[$*. For every* $k \in \mathbb{N}$*, set* $\delta^k = \frac{1}{\tau\gamma} - 1$ *and suppose that* (5.22) *holds. Let* $\left\{\Lambda^k\right\}$ *be sequences of operators satisfying*

$$\begin{cases} \mu \mathrm{Id} \succeq \Lambda^k \succeq \alpha \mathrm{Id}, \\ \mu,\ \alpha \in [\epsilon, 1], \end{cases} \tag{5.36}$$

and let $\left\{\mathbf{x}^k\right\}$ *be a sequence generated by Algorithm 22. Then* $\mathbf{x}^k$ *converges weakly to a solution of Problem* (5.29)*.*

Proof. We defer all proofs to Section 5.4. □

5.3. Connections between ADMM and a second-order primal–dual algorithm when solving ℓ^2+regularizer minimization problems

Many problems in signal processing and machine learning can be addressed by solving a composite minimization problem of the form

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{minimize}} \quad f(\mathbf{x}) + \psi(\mathbf{D}\mathbf{x}), \tag{5.37}$$

where $f : \Gamma(\mathbb{R}^n)$ is a smooth function, $\psi : \Gamma(\mathbb{R}^m)$ is a possibly non-smooth function, and $\mathbf{D} \in \mathbb{R}^{m\times n}$.

ADMM can be used to solve problems of the form of Problem (5.37), as is detailed in Subsection 2.3.1. In that subsection, we used the so-called "scaled" version of ADMM. The difference between the scaled and non-scaled versions is the use of the scaled dual variable instead of the standard dual variable. The non-scaled version of ADMM to solve Problem (5.37) is given by

Algorithm 23: ADMM.

```
Choose $\mathbf{u}^0 \in \mathbb{R}^m,\ \mathbf{d}^0 \in \mathbb{R}^m$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\gamma^k > 0$;
    $\mathbf{x}^{k+1} \leftarrow \arg\min_{\mathbf{x}} \quad L_{\gamma^k}(\mathbf{x}, \mathbf{u}^k, \mathbf{d}^k)$;
    $\mathbf{u}^{k+1} \leftarrow \arg\min_{\mathbf{u}} \quad L_{\gamma^k}(\mathbf{x}^{k+1}, \mathbf{u}, \mathbf{d}^k)$;
    $\mathbf{d}^{k+1} \leftarrow \mathbf{d}^k + \gamma \left(\mathbf{D}\mathbf{x}^{k+1} - \mathbf{u}^{k+1}\right)$;
    $k \leftarrow k + 1$;
end
```

where $L_\gamma(\mathbf{x}, \mathbf{u}, \mathbf{d})$ is

$$L_\gamma(\mathbf{x}, \mathbf{u}, \mathbf{d}) \triangleq f(\mathbf{x}) + \psi(\mathbf{u}) + \frac{\gamma}{2} \left\| \mathbf{D}\mathbf{x} - \mathbf{u} + \frac{\mathbf{d}}{\gamma} \right\|^2 \tag{5.38}$$

and $\gamma > 0$ is a penalization parameter. In this version, $\mathbf{d} \in \mathbb{R}^m$ is the non-scaled version of the dual variable.

It can be shown that, in general, ADMM is able to find solutions at a sublinear rate. Under certain assumptions on one of the functions (e.g., strong convexity and smoothness) and on $\mathbf{D}$ (to be full rank), the rate is linear [66, 53]. In practice, however, one often finds that ADMM is able to solve many problems of practical interest much faster than other first-order methods. This raises the issue of whether ADMM actually makes use of second-order information, at least in some situations. In what follows, we investigate the similarity between ADMM and a variable-metric primal–dual method.

5.3.1. A variable-metric primal–dual method

Condat [67] proposed two different, albeit similar, primal–dual algorithms to solve composite convex problems of the form of Problem (2.3). A variable-metric version of one of them, Algorithm 9, was analyzed by Combettes and Vũ [81]. In this subsection, we discuss a variable-metric version of the other one. To the best of our knowledge, the convergence of this version has not been established in the literature, although it can be proved by a simple variation on the proof devised by Combettes and Vũ. The algorithm is very similar to Algorithm 9 and is given by

Algorithm 24: Forward-backward-based variable-metric primal–dual method (2).

```
Choose $\mathbf{x}^0 \in \mathbb{R}^n,\ \mathbf{d}_1^0 \in \mathbb{R}^{m_1},\ \ldots,\ \mathbf{d}_N^0 \in \mathbb{R}^{m_N}$;
$k \leftarrow 1$;
while stopping criterion is not satisfied do
    Choose $\lambda^k > 0,\ \mathbf{U}^k \succ 0$;
    for $j \leftarrow 1,\ \ldots,\ N$ do
        Choose $\mathbf{U}_j^k \succ 0$;
        $\mathbf{q}_j^k \leftarrow \operatorname{prox}_{h_j^*}^{(\mathbf{U}_j^k)^{-1}} \left(\mathbf{d}_j^k + \mathbf{U}_j^k \mathbf{D}_j \mathbf{x}^k\right)$;
        $\mathbf{d}_j^{k+1} \leftarrow \mathbf{d}_j^k + \lambda^k \left(\mathbf{q}_j^k - \mathbf{d}_j^k\right)$;
    end
    $\mathbf{p}^k \leftarrow \operatorname{prox}_g^{(\mathbf{U}^k)^{-1}} \left(\mathbf{x}^k - \mathbf{U}^k \left(\nabla f\left(\mathbf{x}^k\right) + \sum_{j=1}^N \mathbf{D}_j^* \left(2\mathbf{q}_j^k - \mathbf{d}_j^k\right)\right)\right)$;
    $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \lambda^k \left(\mathbf{p}^k - \mathbf{x}^k\right)$;
    $k \leftarrow k + 1$;
end
```

The following proposition establishes some convergence properties of this algorithm.

Proposition 5.3.1. *Suppose that*

$$0 \in \operatorname{ran}\left(\partial g + \sum_{j=1}^N \mathbf{D}_j^* \partial h_j \mathbf{D}_j + \nabla f\right). \tag{5.39}$$

Let f have a μ^{-1}-Lipschitzian gradient, where $\mu \in \left]0, +\infty\right[$, and let $\epsilon \in \left]0, \min\{1, \mu\}\right[$. For every $k \in \mathbb{N}$, let $\lambda^k \in [\epsilon, 1]$, set (5.21) *and suppose that* (5.22) *holds. Let $\left\{\mathbf{x}^k\right\}$ be a sequence generated by Algorithm 24. Then the following hold:*

1) $\mathbf{x}^k$ *converges weakly to a solution of Problem* (2.3).

2) $\left(\mathbf{d}_1^k, \ldots, \mathbf{d}_N^k\right)$ *converges weakly to a solution of the dual of Problem* (2.3).

Proof. We defer all proofs to Section 5.4. □

Remark 5.3.2. If $h : \mathbf{x} \to 0$, $\mathbf{D}_j = \mathbf{0}, \forall j$ (where $\mathbf{0}$ is a matrix of zeros), $\lambda^k = 1, \forall k$, and $\mathbf{U}^k$ is the Hessian of f, we recover the proximal Newton method [69, 70], which has a quadratic local convergence rate if f is strongly convex.

5.3.2. ℓ^2+regularizer minimization problems

Consider the case when the function f in Problem (5.37) is quadratic, i.e., is of the form $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^*\mathbf{H}^*\mathbf{H}\mathbf{x} + \mathbf{g}^*\mathbf{x} + d$, where $\mathbf{H} \in \mathbb{R}^{m \times n}$, $\mathbf{g} \in \mathbb{R}^n$ and $d \in \mathbb{R}$. This case is very frequent in optimization problems whose objective function is comprised of a quadratic data-fitting term and a regularizer. Such optimization problems are sometimes called ℓ^2+regularizer minimization problems. Since f is convex, $\nabla^2 f(\mathbf{x}) = \mathbf{H}^*\mathbf{H} \succeq 0$.

In what follows, we show how one can apply both ADMM and the variable-metric primal-dual method (VMPD) to solve ℓ^2+regularizer minimization problems. We make two assumptions: (a) that $\mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{D}) = \{\mathbf{0}\}$, where $\mathcal{N}(\cdot)$ denotes the null space of a matrix, and (b) that Problem (5.37) has a unique minimizer. When applying ADMM (cf. Algorithm 23) to solve these problems, Line 5 becomes

$$\left(\mathbf{H}^*\mathbf{H} + \gamma \mathbf{D}^*\mathbf{D}\right)\mathbf{x}^{k+1} = \mathbf{D}^*\left(\gamma \mathbf{u}^k - \mathbf{d}^k\right) - \mathbf{g} \tag{5.40}$$

and Lines 5-7 can then be rewritten as

$$\mathbf{x}^{k+1} = \left(\mathbf{H}^*\mathbf{H} + \gamma \mathbf{D}^*\mathbf{D}\right)^{-1}\left(\mathbf{D}^*\left(\gamma \mathbf{u}^k - \mathbf{d}^k\right) - \mathbf{g}\right), \tag{5.41}$$

$$\mathbf{u}^{k+1} = \operatorname{prox}_{\frac{\psi}{\gamma}}\left(\mathbf{D}\mathbf{x}^{k+1} + \frac{\mathbf{d}^k}{\gamma}\right), \tag{5.42}$$

$$\mathbf{d}^{k+1} = \mathbf{d}^k + \gamma\left(\mathbf{D}\mathbf{x}^{k+1} - \mathbf{u}^{k+1}\right). \tag{5.43}$$

Note that assumption (a), above, ensures that the matrix $(\mathbf{H}^*\mathbf{H} + \gamma \mathbf{D}^*\mathbf{D})$ is nonsingular. ADMM has been used to solve ℓ^2+regularizer problems with great success, which is partly due to how fast this method is at finding solutions to such problems, in practice. This has been hypothesized to be a consequence of the indirect use of second-order information about Problem (5.37) in a way that resembles the use of this type of information in proximal Newton methods [141]. In fact, as can be seen in Eq. (5.41), each iteration of ADMM requires one to solve a linear system whose matrix contains the Hessian of f, $\nabla^2 f(\mathbf{x}) = \mathbf{H}^*\mathbf{H}$. The matrix $(\mathbf{H}^*\mathbf{H} + \gamma \mathbf{D}^*\mathbf{D})$ can be considered to be a modified Hessian.

We now show how to apply VMPD to solve ℓ^2+regularizer minimization problems. It is straightforward to show that the gradient of f is Lipschitz continuous with constant $\lambda_M(\mathbf{H}^*\mathbf{H})$, where $\lambda_M(\mathbf{A})$ denotes the maximum eigenvalue of matrix $\mathbf{A}$. In order to solve Problem (5.37) using Algorithm 24, we make $g : \mathbf{x} \to 0$ and $N = 1$. We also make $\mathbf{U}^k = \nabla^2 f(\mathbf{x}) = \mathbf{H}^*\mathbf{H}$, $\forall_k$, which implies the use of the Hessian of f as a (fixed) metric in Line 7. In this way, we are explicitly using second-order information about the smooth term f, similarly to what is done in the case of proximal Newton methods (see Section 2.3 for details). One may consider using as metric a regularized form of the Hessian by making $\mathbf{U}^k = \mathbf{H}^*\mathbf{H} + \epsilon \mathrm{Id}$, $\forall_k$, where $\epsilon^k > 0$ is an arbitrarily small scalar. This is convenient if $\mathbf{H}^*\mathbf{H}$ is indefinite.[8] In what follows, we consider that $\mathbf{U}^k = \nabla^2 f(\mathbf{x})$; we discuss the regularized case at the end of this subsection. Additionally, we make $\mathbf{U}_1^k = \rho \mathrm{Id}$, $\forall_k$, where $\rho > 0$. We consider the non-relaxed version of this algorithm by making $\lambda^k = 1$, $\forall_k$. Lines 5-11 of Algorithm 24 can be rewritten as

$$\mathbf{d}^{k+1} = \operatorname{prox}_{\rho\psi^*}\left(\mathbf{d}^k + \rho \mathbf{D}\mathbf{x}^k\right), \tag{5.44}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \left(\mathbf{H}^*\mathbf{H}\right)^{-1}\left(\mathbf{H}^*\mathbf{H}\mathbf{x}^k + \mathbf{g} + \mathbf{D}^*\left(2\mathbf{d}^{k+1} - \mathbf{d}^k\right)\right), \tag{5.45}$$

[8] If f is merely convex, $\mathbf{H}^*\mathbf{H}$ is only guaranteed to be PSD. If f is strongly convex, $\mathbf{H}^*\mathbf{H}$ is PD.

by noting that $\nabla f(\mathbf{x}) = \mathbf{H}^*\mathbf{H}\mathbf{x} + \mathbf{g}$, and by making $\mathbf{d}_1^k = \mathbf{d}^k$ and $\mathbf{d}_1^{k+1} = \mathbf{d}^{k+1}, \forall k$, for notational convenience.

In what follows, we study how both algorithms relate to each other. By using Moreau's decomposition (2.24), and introducing an auxiliary variable $\mathbf{u} \in \mathbb{R}^m$, we can rewrite Eqs. (5.44, 5.45) as

$$\mathbf{u}^{k+1} \triangleq \operatorname{prox}_{\frac{\psi}{\rho}}\left(\frac{\mathbf{d}^k}{\rho} + \mathbf{D}\mathbf{x}^k\right), \tag{5.46}$$

$$\mathbf{d}^{k+1} = \mathbf{d}^k + \rho\mathbf{D}\mathbf{x}^k - \rho\mathbf{u}^{k+1}, \tag{5.47}$$

$$\mathbf{x}^{k+1} = -(\mathbf{H}^*\mathbf{H})^{-1}\left(\mathbf{D}^*\left(2\mathbf{d}^{k+1} - \mathbf{d}^k\right) + \mathbf{g}\right). \tag{5.48}$$

By noting that $2\mathbf{d}^{k+1} = \mathbf{d}^{k+1} + \mathbf{d}^{k+1}$, we can rewrite Eq. (5.48) as

$$\mathbf{x}^{k+1} = -(\mathbf{H}^*\mathbf{H})^{-1}\left(\rho\mathbf{D}^*\mathbf{D}\mathbf{x}^k + \mathbf{D}^*\left(\mathbf{d}^{k+1} - \rho\mathbf{u}^{k+1}\right) + \mathbf{g}\right). \tag{5.49}$$

Given this, note that we can rewrite the loop described by Eqs. (5.46-5.48) as

$$\mathbf{x}^k = (\mathbf{H}^*\mathbf{H})^{-1}\left(-\rho\mathbf{D}^*\mathbf{D}\mathbf{x}^{k-1} + \mathbf{D}^*\left(\rho\mathbf{u}^k - \mathbf{d}^k\right) - \mathbf{g}\right), \tag{5.50}$$

$$\mathbf{u}^{k+1} = \operatorname{prox}_{\frac{\psi}{\rho}}\left(\mathbf{D}\mathbf{x}^k + \frac{\mathbf{d}^k}{\rho}\right), \tag{5.51}$$

$$\mathbf{d}^{k+1} = \mathbf{d}^k + \rho\left(\mathbf{D}\mathbf{x}^k - \mathbf{u}^{k+1}\right). \tag{5.52}$$

Finally, by making $\mathbf{x}^k \to \mathbf{x}^{k+1}$ and $\mathbf{x}^{k-1} \to \mathbf{x}^k$, we can rewrite Eqs. (5.50-5.52) as

$$\mathbf{x}^{k+1} = (\mathbf{H}^*\mathbf{H})^{-1}\left(-\rho\mathbf{D}^*\mathbf{D}\mathbf{x}^k + \mathbf{D}^*\left(\rho\mathbf{u}^k - \mathbf{d}^k\right) - \mathbf{g}\right), \tag{5.53}$$

$$\mathbf{u}^{k+1} = \operatorname{prox}_{\frac{\psi}{\rho}}\left(\mathbf{D}\mathbf{x}^{k+1} + \frac{\mathbf{d}^k}{\rho}\right), \tag{5.54}$$

$$\mathbf{d}^{k+1} = \mathbf{d}^k + \rho\left(\mathbf{D}\mathbf{x}^{k+1} - \mathbf{u}^{k+1}\right). \tag{5.55}$$

Let $\mathbf{x}^*$ denote a solution to Problem (5.37); from assumption (b), above, this solution is unique. Since both algorithms converge in the primal variable, i.e., $\mathbf{x}^k \to \mathbf{x}^*$, we also have that $\left\|\mathbf{x}^{k+1} - \mathbf{x}^k\right\| \to 0$. With this in mind, by rearranging Eq. (5.53) as

$$\mathbf{H}^*\mathbf{H}\mathbf{x}^{k+1} + \rho\mathbf{D}^*\mathbf{D}\mathbf{x}^k = \mathbf{D}^*\left(\rho\mathbf{u}^k - \mathbf{d}^k\right) - \mathbf{g}, \tag{5.56}$$

we verify that the closer VMPD gets to $\mathbf{x}^*$, the more similar this equation is to Eq. (5.41) if one makes $\gamma = \rho$. In this sense, these methods can be said to be the "same" in the limit.

There are four important differences between the two methods:

1. ADMM requires the inversion of the matrix $(\mathbf{H}^*\mathbf{H} + \gamma\mathbf{D}^*\mathbf{D})$, whereas VMPD requires the inversion of the matrix $\mathbf{H}^*\mathbf{H}$. This fact may prove to be convenient when using VMPD to solve optimization problems if operations involving $\mathbf{D}$ are computationally challenging.

2. In order to guarantee that ADMM converges to a solution, its use usually requires the matrix $\mathbf{D}$ to be full-rank if f and ψ are merely convex [61]. This is not the case for VMPD, which makes no assumptions on this matrix.

3. If one makes $\mathbf{U}^k = \mathbf{H}^*\mathbf{H} + \epsilon^k \mathrm{Id}$ in case the Hessian is indefinite, ADMM and VMPD are the "same" in the limit only if $\epsilon^k \to 0$. If this condition is not verified, the methods are not exactly the "same" in the limit.

4. The ranges of possible values of the parameters γ and ρ for which the methods converge are not the same. In fact, the only restriction on γ for ADMM to converge is that this parameter should be positive. Moreover, different choices for the values of these parameters have a strong impact on how fast both methods are at finding solutions in practice. Consequently, it is important to choose the values of these parameters that correspond to the fastest versions of the methods. In Subsection 5.3.3, we provide some experimental evidence that both methods exhibit similar speed of convergence if the values of γ and ρ are the same. The value of the parameter γ needs to be hand-tuned, and it is not obvious which values correspond to faster speeds of convergence for ADMM. Usually, the values of these parameters are tuned via simple heuristics, appropriate to the problem at hand, or vary in each iteration following simple strategies [91, 142]. We verified experimentally that the parameter ρ is easier to set, since its range of possible values is bounded and, if one picks its value to be close to one of the bounds, this usually corresponds to the maximum speed of convergence for VMPD.

5.3.3. Experimental study

In this subsection, we experimentally show how ADMM and VMPD compare to each other in the problem of image deblurring. We formulate it as an optimization problem of the form of (1.9). The objective function of this problem is comprised of a quadratic data-fitting term and a regularizer. We consider the regularizer to be isotropic total-variation (Eq. (1.13) with $p = 2$). Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be vectors in $\mathbb{R}^n$, and let $g : \mathbb{R}^n \to]-\infty, +\infty] : (\mathbf{x}_1, \mathbf{x}_2) \to \sum_{i=1}^{n} \|([\mathbf{x}_1]_i, [\mathbf{x}_2]_i)\|_2$. The problem under study is a particular case of Problem (5.37) if one makes $f = \frac{1}{2}\|\mathbf{H}\cdot - \mathbf{y}\|^2$, $\psi = \lambda g$, and $\mathbf{D} = [\mathbf{D}_h^T\ \mathbf{D}_v^T]^T$, where $\mathbf{H} \in \mathbb{R}^{n\times n}$ is a BCCB convolution matrix, and $\mathbf{D}_h \in \mathbb{R}^{n\times n}$ and $\mathbf{D}_v \in \mathbb{R}^{n\times n}$ are such that the products by these matrices compute, respectively, the horizontal and vertical first-order differences of a discrete image, with periodic boundaries. Note that $\mathbf{D}$ is not full column rank and, consequently, ADMM may not converge. If we wished to have a formal guarantee of convergence for this method, we could have modified the discrete difference matrices, making them full rank, e.g., by adding to them $\varepsilon\mathbf{I}_{k+d}$ with a small $\varepsilon > 0$.

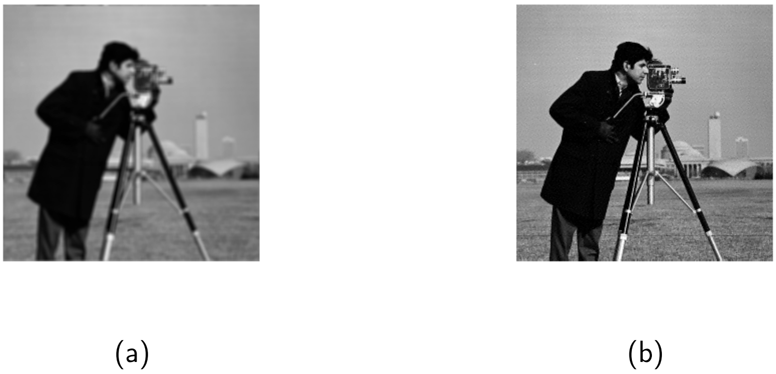


Figure 5.3.: Observed (left) and estimated (right) images using VMPD.

We ran the experiments on the well-known *cameraman* image with size 256×256 pixels, blurred with a low-pass filter. This filter was generated as follows: (a) we generated a truncated Gaussian filter with support of size 5×5 and standard deviation $\sqrt{5}$, (b) we added 1×10^{-2} to the filter coefficient with the highest value, and (c) we normalized the resulting filter, so that after the normalization its largest coefficient was 1. We created the blurred image by performing a circular convolution of the sharp image with the aforementioned filter. We then normalized the image and added i.i.d. Gaussian noise with a BSNR of 50 dB. We used the modified version of the Hessian in VMPD by setting $\epsilon = 10^{-3}$, and we fixed $\gamma = \rho = 2 \times 10^{-6}$, a value that yielded visually good estimated images. The experiments were performed using MATLAB on an Intel Core i7 CPU running at 3.20 GHz, with 32 GB of RAM.

We considered it quite probable that both methods would yield virtually identical estimated images; see Section 3.3.1 for an explanation. In fact, the results for ADMM and VMPD were essentially the same, in the sense that they were visually indistinguishable. For the same reasons given in Section 3.3.1, we arbitrarily chose the result of ADMM after 10^6 iterations as representative of the solution of Problem (5.37).

Figure 5.3 shows the result of VMPD. Fig. 5.4 illustrates the behaviors of both methods during the optimization, which are very similar.

5.4. Proofs

This section includes the proofs of all the propositions, theorems, and corollaries of this chapter. It starts with a preliminary result.

Preliminary result

Lemma 5.4.1. *Let $\alpha \in \left]0, +\infty\right[$ and let $V \in \mathcal{P}_\alpha(\mathcal{X})$. Then*

$$\begin{aligned}\|Vx + (\mathrm{Id} - V)\, y\|^2 = {}& \langle V\,(V - \mathrm{Id})\,(x - y), x - y\rangle \\ & + \|x\|_V^2 - \|y\|_V^2 + \|y\|^2, \quad \forall x \in \mathcal{X}, \quad \forall y \in \mathcal{X}.\end{aligned}$$

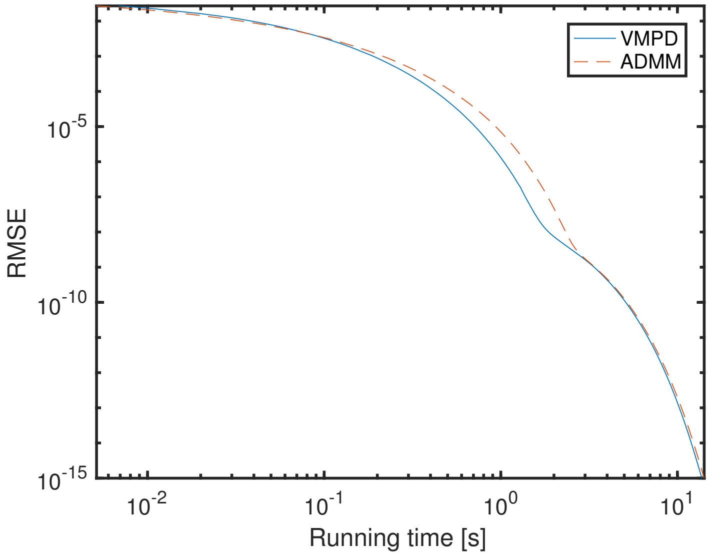


Figure 5.4.: RMSE of the estimated images as a function of running time, for both methods.

Proof. Fix x and y in $\mathcal{X}$. Then

$$\begin{aligned}
\|Vx + (\mathrm{Id} - V)\,y\|^2 &= \langle Vx, Vx\rangle + 2\,\langle Vx, (\mathrm{Id} - V)\,y\rangle + \langle(\mathrm{Id} - V)\,y, (\mathrm{Id} - V)\,y\rangle \\
&= \langle Vx, Vx\rangle + 2\,\langle Vx, y\rangle - 2\,\langle Vx, Vy\rangle + \langle y, y\rangle - 2\,\langle Vy, y\rangle \\
&\quad + \langle Vy, Vy\rangle \\
&\overset{(i)}{=} \langle Vx, Vx\rangle - 2\,\langle Vx, Vy\rangle + \langle Vy, Vy\rangle + \langle Vx, x\rangle \\
&\quad + \langle Vy, y\rangle - \langle V\,(x-y), x-y\rangle + \langle y, y\rangle - 2\,\langle Vy, y\rangle \\
&= \langle V\,(x-y), V\,(x-y)\rangle + \langle Vx, x\rangle - \langle Vy, y\rangle - \langle V\,(x-y), x-y\rangle \\
&\quad + \langle y, y\rangle \\
&= \langle V\,(x-y), (V - \mathrm{Id})\,(x-y)\rangle + \langle Vx, x\rangle - \langle Vy, y\rangle + \langle y, y\rangle\,,
\end{aligned} \tag{5.57}$$

where step (i) follows from the identity $\langle V\,(x-y), x-y\rangle = \langle Vx, x\rangle - 2\,\langle Vx, y\rangle + \langle Vy, y\rangle$. □

Proof of Proposition 5.2.2

Fix x and y in D. By making $V \triangleq \Lambda^{-1}$, (5.5) yields

$$\alpha^{-1}\mathrm{Id} \succeq V \succeq \mu^{-1}\mathrm{Id}. \tag{5.58}$$

By noting that $R = (\mathrm{Id} - V) + VT_\Lambda$, we verify that

$$\begin{aligned}
\|Rx - Ry\|^2 &= \|(\mathrm{Id} - V)(x - y) + V(T_\Lambda x - T_\Lambda y)\|^2 \\
&\overset{(i)}{=} \langle V(V - \mathrm{Id})((T_\Lambda - \mathrm{Id})x - (T_\Lambda - \mathrm{Id})y), (T_\Lambda - \mathrm{Id})x - (T_\Lambda - \mathrm{Id})y\rangle \\
&\quad + \|T_\Lambda x - T_\Lambda y\|_V^2 - \|x - y\|_V^2 + \|x - y\|^2 \\
&\overset{(ii)}{=} \|(T_\Lambda - \mathrm{Id})x - (T_\Lambda - \mathrm{Id})y\|_{V(V-\mathrm{Id})}^2 \\
&\quad + \|T_\Lambda x - T_\Lambda y\|_V^2 - \|x - y\|_V^2 + \|x - y\|^2, \qquad (5.59)
\end{aligned}$$

where step (i) follows from Lemma 5.4.1 and step (ii) follows from the fact that $V(V - \mathrm{Id}) \in \mathcal{P}_0(\mathcal{X})$, since $\alpha^{-1} \geq \mu^{-1} > 1$. The nonexpansiveness of R implies that

$$\begin{aligned}
0 \geq \|Rx - Ry\|^2 - \|x - y\|^2 = \|(T_\Lambda - \mathrm{Id})x - (T_\Lambda - \mathrm{Id})y\|_{V(V-\mathrm{Id})}^2 \\
+ \|T_\Lambda x - T_\Lambda y\|_V^2 - \|x - y\|_V^2. \qquad (5.60)
\end{aligned}$$

Consequently,

$$\|T_\Lambda x - T_\Lambda y\|_V^2 \leq \|x - y\|_V^2 - \|(T_\Lambda - \mathrm{Id})x - (T_\Lambda - \mathrm{Id})y\|_{V(V-\mathrm{Id})}^2. \qquad (5.61)$$

Since, for any given $z \in \mathcal{X}$, $\|z\|_{V(V-\mathrm{Id})}^2 \geq \left(\mu^{-1} - 1\right)\|z\|_V^2$,

$$\|T_\Lambda x - T_\Lambda y\|_V^2 \leq \|x - y\|_V^2 - \left(\mu^{-1} - 1\right)\|(\mathrm{Id} - T_\Lambda)x - (\mathrm{Id} - T_\Lambda)y\|_V^2. \qquad (5.62)$$

The claim follows by noting that $\left(\mu^{-1} - 1\right) = \frac{1-\mu}{\mu}$.

Proof of Theorem 5.2.4

1) Straightforward.

2) Since $x^0 \in D$ and D is convex, Algorithm 17 produces a well-defined sequence in D. By making $V^k \triangleq \left(\Lambda^k\right)^{-1}, \forall k$, (5.7) and [123, Lemma 2.1] yield, for all k,

$$\begin{cases} \left(\alpha^k\right)^{-1}\mathrm{Id} \succeq V^{k+1} \succeq \left(\mu^k\right)^{-1}\mathrm{Id}, \\ (1 + \eta^k)V^k \succeq V^{k+1}. \end{cases} \qquad (5.63)$$

Line 5 of Algorithm 17 implies that, for all k,

$$\begin{aligned}
\left\|x^{k+1} - x^*\right\|_{V^k}^2 &= \left\|T_{\Lambda^k}x^k - T_{\Lambda^k}x^*\right\|_{V^k}^2 \\
&\overset{(i)}{\leq} \left\|x^k - x^*\right\|_{V^k}^2 - \frac{1 - \mu^k}{\mu^k}\left\|(\mathrm{Id} - T_{\Lambda^k})x^k + (\mathrm{Id} - T_{\Lambda^k})x^*\right\|_{V^k}^2 \\
&= \left\|x^k - x^*\right\|_{V^k}^2 - \frac{1 - \mu^k}{\mu^k}\left\|x^{k+1} - x^k\right\|_{V^k}^2 \qquad (5.64) \\
&\leq \left\|x^k - x^*\right\|_{V^k}^2, \qquad (5.65)
\end{aligned}$$

where step (i) follows from (5.63) and Proposition 5.2.2.

Since, for any given $z \in \mathcal{X}$, we verify from (5.63) that

$$(1+\eta^k)\,\|z\|^2_{V^k} \geq \|z\|^2_{V^{k+1}}\,, \quad \forall k \in \mathbb{N}, \tag{5.66}$$

using (5.65), we get

$$\begin{aligned}\left\|x^{k+1}-x^*\right\|^2_{V^{k+1}} &\leq (1+\eta^k)\left\|x^{k+1}-x^*\right\|^2_{V^k} \\ &\leq (1+\eta^k)\left\|x^k-x^*\right\|^2_{V^k}.\end{aligned} \tag{5.67}$$

3) Since $\{x^k\}$ is Fejér monotone with respect to Fix R relative to $\{V^k\}$, the sequence $\left\{\|x^k-x^*\|^2_{V^k}\right\}$ converges [123, Proposition 3.2(i)]. Define

$$\zeta \triangleq \sup_k \left\|x^k-x^*\right\|_{V^k} < +\infty. \tag{5.68}$$

It follows from Line 5 of Algorithm 17 and (5.7) that, for all k,

$$\begin{aligned}\left\|x^{k+1}-x^k\right\|^2_{V^k} &= \left\|\Lambda^k\left(Rx^k-x^k\right)\right\|^2_{V^k} \\ &= \left\|\left(\Lambda^k\right)^{\frac{1}{2}}\left(Rx^k-x^k\right)\right\|^2 \\ &\geq \alpha^k\left\|Rx^k-x^k\right\|^2.\end{aligned} \tag{5.69}$$

and, in view of (5.64), we can rewrite (5.67) as

$$\begin{aligned}\left\|x^{k+1}-x^*\right\|^2_{V^{k+1}} &\leq (1+\eta^k)\left(\left\|x^k-x^*\right\|^2_{V^k} - \frac{1-\mu^k}{\mu^k}\left\|x^{k+1}-x^k\right\|^2_{V^k}\right) \\ &\leq (1+\eta^k)\left\|x^k-x^*\right\|^2_{V^k} - \left(1-\mu^k\right)\left\|x^{k+1}-x^k\right\|^2_{V^k} \\ &\leq (1+\eta^k)\left\|x^k-x^*\right\|^2_{V^k} - \alpha^k(1-\mu^k)\left\|Rx^k-x^k\right\|^2 \\ &\leq \left\|x^k-x^*\right\|^2_{V^k} + \zeta^2\eta^k - \epsilon^2\left\|Rx^k-x^k\right\|^2.\end{aligned} \tag{5.70}$$

For every $K \in \mathbb{N}$, by iterating (5.70) we can write that

$$\begin{aligned}\epsilon^2\sum_{k=0}^{K}\left\|Rx^k-x^k\right\|^2 &\leq \|x^0-x^*\|^2_{V^0} - \|x^{K+1}-x^*\|^2_{V^{K+1}} + \sum_{k=0}^{K}\zeta^2\eta^k \\ &\leq \zeta^2 + \sum_{k=0}^{K}\zeta^2\eta^k.\end{aligned} \tag{5.71}$$

Since $\{\eta^k\}$ is absolutely summable, taking the limit as $K \to +\infty$ yields

$$\sum_{k=0}^{\infty} \left\| Rx^k - x^k \right\|^2 \leq \frac{1}{\epsilon^2} \left(\zeta^2 + \sum_{k=0}^{\infty} \zeta^2 \eta^k \right) < \infty. \tag{5.72}$$

Consequently, $Rx^k - x^k \to 0$.

4) Let x be a weak sequential cluster point of $\{x^k\}$. It follows from [11, Corollary 4.18] that $x \in \operatorname{Fix} R$. In view of [123, Lemma 2.3] and [123, Theorem 3.3], the proof is complete.

Proof of Theorem 5.2.5

We start by defining, for all k,

$$\begin{cases} A^k \triangleq \gamma^k U^k A, \\ C^k \triangleq \gamma^k U^k C, \\ \Phi^k \triangleq U^k \Lambda^k, \end{cases} \quad \text{and} \quad \begin{cases} p^k \triangleq J_{A^k} y^k, \\ q^k \triangleq J_{A^k} \left(x^k - C^k x^k \right), \\ s^k \triangleq x^k + \Lambda^k \left(q^k - x^k \right) \end{cases} \tag{5.73}$$

and by recalling some results derived by Combettes and Vũ [81]. We have from [81, Eq. (4.8)] that

$$\left\| p^k - q^k \right\|_{(U^k)^{-1}} \leq \frac{2\beta - \epsilon}{\sqrt{\mu_U}} \left\| b^k \right\|. \tag{5.74}$$

Additionally, for any $x^* \in Z$, from [81, Eq. (4.12)] we can write that

$$\begin{aligned} \left\| q^k - x^* \right\|^2_{(U^k)^{-1}} \leq & \left\| x^k - x^* \right\|^2_{(U^k)^{-1}} - \epsilon^2 \left\| Cx^k - Cx^* \right\|^2 \\ & - \left\| (x^k - q^k) - (C^k x^k - C^k x^*) \right\|^2_{(U^k)^{-1}}. \end{aligned} \tag{5.75}$$

We now establish some identities. Line 6 of Algorithm 18 and (5.73) imply, for all k, that

$$
\begin{aligned}
\left\| x^{k+1} - s^k \right\|_{(\Phi^k)^{-1}} &= \left\| \left(x^k + \Lambda^k \left(p^k + a^k - x^k \right) \right) - \left(x^k + \Lambda^k \left(q^k - x^k \right) \right) \right\|_{(\Phi^k)^{-1}} \\
&= \left\| \Lambda^k \left(p^k + a^k - q^k \right) \right\|_{(\Phi^k)^{-1}} \\
&\le \left\| \Lambda^k a^k \right\|_{(\Phi^k)^{-1}} + \left\| \Lambda^k \left(p^k - q^k \right) \right\|_{(\Phi^k)^{-1}} \\
&= \left\| \sqrt{(U^k)^{-1}} \sqrt{\Lambda^k} a^k \right\| + \left\| \sqrt{\Lambda^k} \left(p^k - q^k \right) \right\|_{(U^k)^{-1}} \\
&\le \left\| \sqrt{(U^k)^{-1}} \right\| \left\| \sqrt{\Lambda^k} \right\| \left\| a^k \right\| + \left\| \sqrt{\Lambda^k} \right\|_{(U^k)^{-1}} \left\| p^k - q^k \right\|_{(U^k)^{-1}} \\
&\le \sqrt{\left\| (U^k)^{-1} \right\|} \sqrt{\| \Lambda^k \|} \left\| a^k \right\| + \sqrt{\left\| (U^k)^{-1} \right\|} \sqrt{\| \Lambda^k \|} \left\| p^k - q^k \right\|_{(U^k)^{-1}} \\
&\overset{(5.74)}{\le} \sqrt{\mu} \left(\frac{1}{\sqrt{\alpha_U}} \left\| a^k \right\| + \frac{2\beta - \epsilon}{\sqrt{\alpha_U \mu_U}} \left\| b^k \right\| \right) \\
&\le \frac{1}{\sqrt{\alpha_U}} \left\| a^k \right\| + \frac{2\beta - \epsilon}{\sqrt{\alpha_U \mu_U}} \left\| b^k \right\|
\end{aligned} \tag{5.76}
$$

and that

$$
\begin{aligned}
\left\| s^k - x^* \right\|^2_{(\Phi^k)^{-1}} &= \left\| \left(x^k + \Lambda^k \left(q^k - x^k \right) \right) - x^* \right\|^2_{(\Phi^k)^{-1}} \\
&= \left\| \left(\mathrm{Id} - \Lambda^k \right) \left(x^k - x^* \right) + \Lambda^k (q^k - x^*) \right\|^2_{(\Phi^k)^{-1}} \\
&\overset{(i)}{=} \left\langle \Lambda^k \left(\Lambda^k - \mathrm{Id} \right) \left(q^k - x^k \right), q^k - x^k \right\rangle_{(\Phi^k)^{-1}} \\
&\quad + \left\| q^k - x^* \right\|^2_{(\Phi^k)^{-1}\Lambda^k} - \left\| x^k - x^* \right\|^2_{(\Phi^k)^{-1}\Lambda^k} + \left\| x^k - x^* \right\|^2_{(\Phi^k)^{-1}} \\
&= - \left\| q^k - x^k \right\|^2_{(U^k)^{-1}(\mathrm{Id} - \Lambda^k)} + \left\| q^k - x^* \right\|^2_{(U^k)^{-1}} - \left\| x^k - x^* \right\|^2_{(U^k)^{-1}} \\
&\quad + \left\| x^k - x^* \right\|^2_{(\Phi^k)^{-1}} \\
&\le \left\| q^k - x^* \right\|^2_{(U^k)^{-1}} - \left\| x^k - x^* \right\|^2_{(U^k)^{-1}} + \left\| x^k - x^* \right\|^2_{(\Phi^k)^{-1}} \\
&\overset{(ii)}{\le} -\epsilon^2 \left\| C x^k - C x^* \right\|^2 - \left\| (x^k - q^k) - (C^k x^k - C^k x^*) \right\|^2_{(U^k)^{-1}} \\
&\quad + \left\| x^k - x^* \right\|^2_{(\Phi^k)^{-1}}
\end{aligned} \tag{5.77}
$$

where step (i) follows from Lemma 5.4.1 and step (ii) from inequation (5.75).

Since, for any given $z \in \mathcal{X}$ and for all k, from [123, Lemma 2.1] we verify that

$$
(1 + \eta^k)(\Phi^k)^{-1} \succeq (\Phi^{k+1})^{-1} \quad \text{and} \quad (1 + \eta^k) \|z\|^2_{(\Phi^k)^{-1}} \ge \|z\|^2_{(\Phi^{k+1})^{-1}}, \tag{5.78}
$$

using inequation (5.77), we get

$$\begin{aligned}\left\|s^k - x^*\right\|^2_{(\Phi^{k+1})^{-1}} &\leq (1+\eta^k)\left\|x^k - x^*\right\|^2_{(\Phi^k)^{-1}} \\ &\quad - \epsilon^2\left\|C^k x - Cx^*\right\|^2 - \left\|(x^k - q^k) - (C^k x^k - C^k x^*)\right\|^2_{(U^k)^{-1}} \quad (5.79)\\ &\leq (1+\eta^k)\left\|x^k - x^*\right\|^2_{(\Phi^k)^{-1}} \quad (5.80)\\ &\leq \delta^2\left\|x^k - x^*\right\|^2_{(\Phi^k)^{-1}}, \quad (5.81)\end{aligned}$$

where $\delta \triangleq \sup_k \sqrt{1+\eta^k}$.

We also define

$$\epsilon^k \triangleq \delta\left(\frac{1}{\sqrt{\alpha_U}}\left\|a^k\right\| + \frac{2\beta - \epsilon}{\sqrt{\alpha_U \mu_U}}\left\|b^k\right\|\right). \quad (5.82)$$

Finally, inequalities (5.76), (5.78) and (5.82), yield

$$\begin{aligned}\left\|x^{k+1} - s^k\right\|^2_{(\Phi^{k+1})^{-1}} &\leq (1+\eta^k)\left\|x^{k+1} - s^k\right\|^2_{(\Phi^k)^{-1}} \quad (5.83)\\ &\leq (\epsilon^k)^2. \quad (5.84)\end{aligned}$$

1) We are now able to prove quasi-Fejér monotonicity of $\left\{x^k\right\}$:

$$\begin{aligned}\left\|x^{k+1} - x^*\right\|_{(\Phi^{k+1})^{-1}} &\leq \left\|x^{k+1} - s^k\right\|_{(\Phi^{k+1})^{-1}} + \left\|s^k - x^*\right\|_{(\Phi^{k+1})^{-1}}\\ &\overset{(i)}{\leq} \sqrt{1+\eta^k}\left\|x^{k+1} - s^k\right\|_{(\Phi^k)^{-1}} + \sqrt{1+\eta^k}\left\|x^k - x^*\right\|_{(\Phi^k)^{-1}}\\ &\overset{(ii)}{\leq} \epsilon^k + \sqrt{1+\eta^k}\left\|x^k - x^*\right\|_{(\Phi^k)^{-1}}\\ &\leq (1+\eta^k)\left\|x^k - x^*\right\|_{(\Phi^k)^{-1}} + \epsilon^k, \quad (5.85)\end{aligned}$$

where step (i) follows from inequalities (5.83) and (5.80) and step (ii) follows from inequality (5.83).

Since $\left\{a^k\right\}$ and $\left\{b^k\right\}$ are absolutely summable, $\sum_k \epsilon^k < +\infty$. From the assumptions and from (5.78) and (5.85), in view of [123, Proposition 4.1(i)], we conclude that $\left\{x^k\right\}$ is quasi-Fejér monotone with respect to Z relative to $\left\{(\Phi^k)^{-1}\right\}$.

2) As a consequence of *1)* and [123, Proposition 4.1(ii)], $\left\{\left\|x^k - x^*\right\|_{(\Phi^k)^{-1}}\right\}$ converges. We define

$$\zeta \triangleq \sup_k \left\|x^k - x^*\right\|_{(\Phi^k)^{-1}} < +\infty. \quad (5.86)$$

Moreover,

$$\begin{aligned}
\left\|x^{k+1}-x^*\right\|^2_{(\Phi^{k+1})^{-1}} &= \left\|x^{k+1}-s^k+s^k-x^*\right\|^2_{(\Phi^{k+1})^{-1}} \\
&\leq \left\|s^k-x^*\right\|^2_{(\Phi^{k+1})^{-1}} + 2\left\|s^k-x^*\right\|_{(\Phi^{k+1})^{-1}} \left\|x^{k+1}-s^k\right\|_{(\Phi^{k+1})^{-1}} \\
&\quad + \left\|x^{k+1}-s^k\right\|^2_{(\Phi^{k+1})^{-1}} \\
&\overset{(i)}{\leq} (1+\eta^k)\left\|x^k-x^*\right\|^2_{(\Phi^k)^{-1}} - \epsilon^2\left\|Cx^k-Cx^*\right\|^2 \\
&\quad - \left\|(x^k-q^k)-(C^kx^k-C^kx^*)\right\|^2_{(U^k)^{-1}} + 2\delta\zeta\epsilon^k+(\epsilon^k)^2 \\
&\leq \left\|x^k-x^*\right\|^2_{(\Phi^k)^{-1}} - \epsilon^2\left\|Cx^k-Cx^*\right\|^2 \\
&\quad - \left\|(x^k-q^k)-(C^kx^k-C^kx^*)\right\|^2_{(U^k)^{-1}} + \zeta^2\eta^k + 2\delta\zeta\epsilon^k+(\epsilon^k)^2,
\end{aligned} \tag{5.87}$$

where step (i) follows from (5.79), (5.81), (5.86) and (5.83).

For every $K \in \mathbb{N}$, by iterating (5.87), we can write that

$$\begin{aligned}
\epsilon^2\sum_{k=0}^{K}\left\|Cx^k-Cx^*\right\|^2 &\leq \left\|x^0-x^*\right\|^2_{(\Phi^0)^{-1}} - \left\|x^{I+1}-x^*\right\|^2_{(\Phi^{I+1})^{-1}} + \sum_{k=0}^{K}\left(\zeta^2\eta^k+2\delta\zeta\epsilon^k+(\epsilon^k)^2\right) \\
&\leq \zeta^2 + \sum_{k=0}^{K}\left(\zeta^2\eta^k+2\delta\zeta\epsilon^k+(\epsilon^k)^2\right).
\end{aligned} \tag{5.88}$$

Taking the limit from this inequality as $K \to +\infty$ yields

$$\sum_{k=0}^{\infty}\left\|Cx^k-Cx^*\right\|^2 \leq \frac{1}{\epsilon^2}\left(\zeta^2+\sum_{n}\left(\zeta^2\eta^k+2\delta\zeta\epsilon^k+(\epsilon^k)^2\right)\right) < +\infty, \tag{5.89}$$

since $\{\eta^k\}$ and $\{\epsilon^k\}$ are absolutely summable. Following a similar reasoning, we can show that

$$\sum_{k=0}^{\infty}\left\|(x^k-q^k)-(C^kx^k-C^kx^*)\right\|^2_{(U^k)^{-1}} < +\infty. \tag{5.90}$$

Let x be a weak sequential cluster point of $\{x^k\}$. In view of [123, Theorem 3.3], it remains to be shown that $x \in Z$. Since $\left\{\left\|Cx^k-Cx^*\right\|^2\right\}$ and $\left\{\left\|(x^k-q^k)-(C^kx^k-C^kx^*)\right\|^2_{(U^k)^{-1}}\right\}$ are absolutely summable, using the same arguments as in [81, Eqs. (4.26)-(4.31)], then $-Cx \in Ax$, which is equivalent to $x \in Z$.

Proof of Corollary 5.2.8

By [11, Theorem 20.40], ∂f is maximally monotone and, by [11, Corollary 18.16], ∇g is β-cocoercive. Additionally, Argmin $(f+g) = \text{zer}\,(\partial f + \nabla g)$, by [11, Corollary 26.3]. Then the present corollary is an application of Theorem 5.2.5, by making $A = \partial f$ and $C = \nabla g$.

Proof of Corollary 5.2.10

The proof provided here follows the structure of similar proofs [122, 68, 123] and is organized as follows: (a) we show that Algorithm 20 is an instance of Algorithm 18, and (b) that the assumptions of the present corollary satisfy the convergence conditions of Theorem 5.2.5.

We start by introducing some notation. We denote by $\boldsymbol{\mathcal{V}}$ the Hilbert direct sum of the real Hilbert spaces $\mathcal{V}_{j\in\{1,\dots,N\}}$, i.e., $\boldsymbol{\mathcal{V}} = \bigoplus_{j\in\{1,\dots,N\}} \mathcal{V}_j$. We endow this space with the following scalar product and norm, respectively:

$$\langle\cdot,\cdot\rangle_{\boldsymbol{\mathcal{V}}} : (\boldsymbol{a},\boldsymbol{b}) \mapsto \sum_{j=1}^{N} \omega_j \langle a_j, b_j\rangle \quad \text{and} \quad \|\cdot\|_{\boldsymbol{\mathcal{V}}} : \boldsymbol{a} \mapsto \sqrt{\sum_{j=1}^{N} \omega_j \|a_j\|^2}, \tag{5.91}$$

where $\boldsymbol{a} = (a_1,\dots,a_j), \boldsymbol{b} = (b_1,\dots,b_j) \in \boldsymbol{\mathcal{V}}$. Additionally, we denote by $\boldsymbol{\mathcal{K}}$ the Hilbert direct sum $\boldsymbol{\mathcal{K}} = \mathcal{X} \oplus \boldsymbol{\mathcal{V}}$ and endow the resulting space with the following scalar product and norm, respectively:

$$\langle\cdot,\cdot\rangle_{\boldsymbol{\mathcal{K}}} : \big((x,\mathbf{a}),(y,\mathbf{b})\big) \mapsto \langle x,y\rangle + \langle \mathbf{a},\mathbf{b}\rangle_{\boldsymbol{\mathcal{V}}} \quad \text{and} \quad \|\cdot\|_{\boldsymbol{\mathcal{K}}} : (x,\mathbf{a}) \mapsto \sqrt{\|x\|^2 + \|\mathbf{a}\|_{\boldsymbol{\mathcal{V}}}}, \tag{5.92}$$

where $x,y \in \mathcal{X}$.

We define, for all $k \in \mathbb{N}$,

$$\begin{cases} \boldsymbol{d}^k \in \boldsymbol{\mathcal{V}} \triangleq \left(d_1^k,\dots,d_N^k\right), \\ \boldsymbol{x}^k \in \boldsymbol{\mathcal{K}} \triangleq \left(x^k, \boldsymbol{d}^k\right), \\ \mathbf{y}^k \in \boldsymbol{\mathcal{K}} \triangleq \left(p^k, q_1^k,\dots,q_N^k\right), \\ \boldsymbol{a}^k \in \boldsymbol{\mathcal{K}} \triangleq \left(a^k, b_1^k,\dots,b_N^k\right), \\ \boldsymbol{c}^k \in \boldsymbol{\mathcal{K}} \triangleq \left(c^k, e_1^k,\dots,e_N^k\right), \\ \mathbf{f}^k \in \boldsymbol{\mathcal{K}} \triangleq \left(\left(U^k\right)^{-1} a^k, \left(U_1^k\right)^{-1} b_1^k,\dots,\left(U_N^k\right)^{-1} b_N^k\right). \end{cases} \tag{5.93}$$

We also define the operators

$$\begin{cases} \boldsymbol{A} : \boldsymbol{\mathcal{K}} \to 2^{\boldsymbol{\mathcal{K}}} : (x,\mathbf{a}) \to \left(\sum_{j=1}^N \omega_j L_j^* a_j - z + Ax\right) \times \left(-L_1 x + r_1 + {B_1}^{-1} a_1\right) \times \dots \\ \qquad\qquad\qquad\qquad\qquad \times \left(-L_N x + r_N + {B_N}^{-1} a_N\right), \\ \boldsymbol{C} : \boldsymbol{\mathcal{K}} \to \boldsymbol{\mathcal{K}} : (x,\mathbf{a}) \to \left(Cx, E_1^{-1} a_1,\dots, E_N^{-1} a_N\right), \\ \mathbf{S} : \boldsymbol{\mathcal{K}} \to \boldsymbol{\mathcal{K}} : (x,\mathbf{a}) \to \left(\sum_{j=1}^N \omega_j L_j^* a_j, -L_1 x,\dots,-L_N x\right), \end{cases} \tag{5.94}$$

and[9]

$$\begin{cases} \boldsymbol{U}^k : \boldsymbol{\mathcal{K}} \to \boldsymbol{\mathcal{K}} : (x, \mathbf{a}) \to \left(U^k x, U_1^k a_1, \dots, U_N^k a_N\right), \\ \mathbf{V}^k : \boldsymbol{\mathcal{K}} \to \boldsymbol{\mathcal{K}} : (x, \mathbf{a}) \to \left(\left(U^k\right)^{-1} x + \sum_{j=1}^N \omega_j L_j^* a_j, \left(U_1^k\right)^{-1} a_1 + L_1 x, \dots, \left(U_N^k\right)^{-1} a_N + L_N x\right), \\ \mathbf{T}^k : \mathcal{X} \to \boldsymbol{\mathcal{V}} : x \to \left(\sqrt{U_1^k} L_1 x, \dots, \sqrt{U_N^k} L_N x\right), \\ \boldsymbol{\Lambda}^k : \boldsymbol{\mathcal{K}} \to \boldsymbol{\mathcal{K}} : (x, \mathbf{a}) \to \left(\Lambda^k x, \Lambda_1^k a_1, \dots, \Lambda_N^k a_N\right), \\ \boldsymbol{\Phi}^k : \boldsymbol{\mathcal{K}} \to \boldsymbol{\mathcal{K}} : (x, \mathbf{a}) \to \left(\Lambda^k U^k x, \Lambda_1^k U_1^k a_1, \dots, \Lambda_N^k U_N^k a_N\right). \end{cases} \tag{5.95}$$

Algorithm 20 can be written as

Algorithm 25: An alternative formulation of Algorithm 20.

```
1  Choose $x^0 \in \mathcal{X}$ and $d_1^0 \in \mathcal{V}_1, \cdots, d_j^0 \in \mathcal{V}_j$;
2  $k \leftarrow 1$;
3  while stopping criterion is not satisfied do
4      for $j \leftarrow 1, \dots, N$ do
5          Choose $U_j^k \succ 0$, and $\mathrm{Id} \succ \Lambda_j^k \succ 0$;
6          $\left(U_j^k\right)^{-1}\left(d_j^k - q_j^k\right) + L_j x^k - E_j^{-1} d_j^k \in r_j + B_j^{-1}\left(q_j^k - b_j^k\right) + e_j^k - \left(U_j^k\right)^{-1} b_j^k$;
7          $d_j^{k+1} = d_j^k + \Lambda_j^k\left(q_j^k - d_j^k\right)$;
8      end
9      Choose $U^k \succ 0$, and $\mathrm{Id} \succ \Lambda^k \succ 0$;
10     $\left(U^k\right)^{-1}\left(x^k - p^k\right) + \sum_{j=1}^N \omega_j L_j^*\left(d_j^k - q_j^k\right) - Cx^k \in$
           $-z + A\left(p^k - a^k\right) + \sum_{j=1}^N \omega_j L_j^* q_j^k + c^k - \left(U^k\right)^{-1} a^k$;
11     $x^{k+1} = x^k + \Lambda^k\left(p^k - x^k\right)$;
12     $k \leftarrow k + 1$;
13 end
```

Lines 6 and 10 of Algorithm 25 can be rewritten as

$$\mathbf{V}\left(\boldsymbol{x}^k - \mathbf{y}^k\right) - \boldsymbol{C}\boldsymbol{x}^k \in \boldsymbol{A}(\mathbf{y}^k - \boldsymbol{a}^k) + \mathbf{S}\boldsymbol{a}^k + \boldsymbol{c}^k - \mathbf{f}^k, \tag{5.96}$$

whereas lines 7 and 11 can be rewritten as

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{\Lambda}^k\left(\mathbf{y}^k - \boldsymbol{x}^k\right). \tag{5.97}$$

Set, for all k,

$$\mathbf{b}^k \triangleq \left(\mathbf{S} + \mathbf{V}^k\right)\boldsymbol{a}^k + \boldsymbol{c}^k - \mathbf{f}^k. \tag{5.98}$$

[9] Note that the definition of the operator $\mathbf{V}^k$ in (5.95) is not the same as the equivalent operator in [81, Eq. (6.10)].

Using the same arguments as in [68, Eqs. (3.25)-(3.30)], it can be shown that expression (5.96) can be rewritten as

$$\mathbf{y}^k = J_{\left(\mathbf{V}^k\right)^{-1}\boldsymbol{A}}\left(\boldsymbol{x}^k - \left(\mathbf{V}^k\right)^{-1}\left(\boldsymbol{C}\boldsymbol{x}^k + \mathbf{b}^k\right)\right) + \boldsymbol{a}^k. \tag{5.99}$$

In view of Eqs. (5.99) and (5.97), it is clear that Algorithm 20 is an instance of Algorithm 18 if one makes $\gamma^k = 1$ for all k.

We now show that the assumptions of the current corollary satisfy the conditions of Theorem 5.2.5. We start by noting that the operator $\boldsymbol{A}$ is maximally monotone [68, Eqs. (3.7)-(3.9)] and the operator $\boldsymbol{C}$ is β-cocoercive [68, Eq. (3.12)].

Next, we show that the operators $\left(\mathbf{V}^k\right)^{-1}$ and $\boldsymbol{\Lambda}^k$ satisfy assumptions (5.8) and (5.9) for all k. It follows from the assumptions of the present corollary and [81, Lemma 3.1] that

$$\boldsymbol{U}^{k+1} \succeq \boldsymbol{U}^k \in \mathcal{P}_{\alpha_U}(\boldsymbol{\mathcal{K}}) \quad \text{and} \quad \left\|\left(\boldsymbol{U}^k\right)^{-1}\right\| \leq \frac{1}{\alpha_U}. \tag{5.100}$$

Since, for all k, $\boldsymbol{U}^k \in \mathcal{S}(\boldsymbol{\mathcal{K}})$, than $\mathbf{V}^k \in \mathcal{S}(\boldsymbol{\mathcal{K}})$. Additionally, by noting that $\mathbf{S}$ is linear and bounded, we verify, for all k, that

$$\left\|\mathbf{V}^k\right\| \leq \left\|\left(\boldsymbol{U}^k\right)^{-1}\right\| + \|\mathbf{S}\| \leq \frac{1}{\alpha_U} + \sqrt{\sum_{j=1}^{N} \|L_j\|^2} \triangleq \rho \tag{5.101}$$

and that, for every $x \in \mathcal{X}$

$$\begin{aligned}\left\|\mathbf{T}^k x\right\|_{\boldsymbol{\mathcal{V}}}^2 &= \sum_{j=1}^{N} \omega_j \left\|\sqrt{U_j^k} L_j \sqrt{U^k}\left(U^k\right)^{-\frac{1}{2}} x\right\|^2 \\ &\leq \|x\|_{(U^k)^{-1}}^2 \sum_{j=1}^{N} \omega_j \left\|\sqrt{U_j^k} L_j \sqrt{U^k}\right\|^2 \\ &= \beta^k \|x\|_{(U^k)^{-1}}^2,\end{aligned} \tag{5.102}$$

where $\beta^k \triangleq \sum_{j=1}^{N} \omega_j \left\|\sqrt{U_j^k} L_j \sqrt{U^k}\right\|^2, \forall_k$. Then, following the arguments made in [81, Eq. 6.15], for every k and every $\boldsymbol{x} = (x, a_1, \ldots, a_j) \in \boldsymbol{\mathcal{K}}$, we obtain

$$\begin{aligned}
\left\langle \mathbf{V}^k \boldsymbol{x}, \boldsymbol{x} \right\rangle_{\boldsymbol{\mathcal{K}}} &= \left\langle \left(U^k\right)^{-1} x, x \right\rangle + 2 \sum_{j=1}^{N} \omega_j \left\langle L_j x, a_j \right\rangle + \sum_{j=1}^{N} \omega_j \left\langle \left(U_j^k\right)^{-1} a_j, a_j \right\rangle \\
&= \|x\|^2_{(U^k)^{-1}} + \sum_{j=1}^{N} \omega_j \|a_j\|^2_{(U_j^k)^{-1}} + 2 \sum_{j=1}^{N} \omega_j \left\langle \sqrt{U_j^k} L_j x, \left(U_j^k\right)^{-\frac{1}{2}} a_j \right\rangle \\
&= \|x\|^2_{(U^k)^{-1}} + \sum_{j=1}^{N} \omega_j \|a_j\|^2_{(U_j^k)^{-1}} \\
&\quad + 2 \left\langle \left((1+\delta^k)\beta^k\right)^{-\frac{1}{2}} \mathbf{T}^k x, \sqrt{(1+\delta^k)\beta^k} \left(\left(U_1^k\right)^{-\frac{1}{2}} a_1, \ldots, \left(U_N^k\right)^{-\frac{1}{2}} a_N \right) \right\rangle_{\boldsymbol{\mathcal{V}}} \\
&\overset{(i)}{\geq} \|x\|^2_{(U^k)^{-1}} + \sum_{j=1}^{N} \omega_j \|a_j\|^2_{(U_j^k)^{-1}} - \left(\frac{\left\|\mathbf{T}^k x\right\|^2_{\boldsymbol{\mathcal{V}}}}{(1+\delta^k)\beta^k} + (1+\delta^k)\beta^k \sum_{j=1}^{N} \omega_j \|a_j\|_{(U_j^k)^{-1}} \right) \\
&\overset{(ii)}{\geq} \|x\|^2_{(U^k)^{-1}} + \sum_{j=1}^{N} \omega_j \|a_j\|^2_{(U_j^k)^{-1}} - \left(\frac{\|x\|^2_{(U^k)^{-1}}}{1+\delta^k} + (1+\delta^k)\beta^k \sum_{j=1}^{N} \omega_j \|a_j\|_{(U_j^k)^{-1}} \right) \\
&\overset{(iii)}{=} \frac{\delta^k}{1+\delta^k} \left(\|x\|^2_{(U^k)^{-1}} + \sum_{j=1}^{N} \omega_j \|a_j\|^2_{(U_j^k)^{-1}} \right) \\
&\geq \frac{\delta^k}{1+\delta^k} \left(\left\|U^k\right\|^{-1} \|x\|^2 + \sum_{j=1}^{N} \omega_j \left\|U_j^k\right\|^{-1} \|a_j\|^2 \right) \\
&\overset{(iv)}{\geq} \xi^k \|x\|^2_{\boldsymbol{\mathcal{K}}}
\end{aligned}$$

where step (i) follows from the identity $2\langle \mathbf{a}, \mathbf{b} \rangle \geq -\|\mathbf{a}\|^2 - \|\mathbf{b}\|^2$, step (ii) follows from inequality (5.102), step (iii) follows from the fact that $(1+\delta^k)\beta^k = \frac{1}{1+\delta^k}$, and step (iv) follows from assumption (5.22). Following the arguments made in [81, Eqs. (6.16)-(6.18)], this last inequation implies that

$$\sup_k \left\| \left(\mathbf{V}^k\right)^{-1} \right\| \leq 2\beta - \epsilon \quad \text{and} \quad \left(\mathbf{V}^{k+1}\right)^{-1} \succeq \left(\mathbf{V}^k\right)^{-1} \in \mathcal{P}_{1/\rho}(\boldsymbol{\mathcal{K}}), \tag{5.103}$$

which satisfies assumption (5.8) of Theorem 5.2.5. It follows from the assumptions of the present corollary that

$$\sup_k \left\|\mathbf{\Lambda}^k\right\| \leq 1, \quad \mathbf{\Lambda}^{k+1} \succeq \mathbf{\Lambda}^k \in \mathcal{P}_\alpha(\boldsymbol{\mathcal{K}}), \tag{5.104}$$

and

$$\mathbf{\Phi}^{k+1} \succeq \mathbf{\Phi}^k. \tag{5.105}$$

Moreover, it follows from [81, Lemma 3.1] that $\sum_k \|\boldsymbol{a}\|_{\boldsymbol{\mathcal{K}}} \leq +\infty$, $\sum_k \|\boldsymbol{c}\|_{\boldsymbol{\mathcal{K}}} \leq +\infty$, and $\sum_k \|\mathbf{f}\|_{\boldsymbol{\mathcal{K}}} \leq +\infty$. From Eqs. (5.98) and (5.101), it follows that $\sum_k \|\mathbf{b}\|_{\boldsymbol{\mathcal{K}}} \leq +\infty$.

It is shown in [122, Eq. (3.13)] that under assumptions (5.13), zer $(\boldsymbol{A} + \boldsymbol{C}) \neq \emptyset$. Additionally, following the arguments made in [122, Eqs. (3.21) and (3.22)], if $(x^*, \boldsymbol{d}^*) \in$ zer $(\boldsymbol{A}+\boldsymbol{C})$, then $x^* \in P$ and $\boldsymbol{d}^* \in D$. Then, the conditions of Theorem 5.2.5 are satisfied. Consequently, we have a $\boldsymbol{x}^* \triangleq \left(x^*, d_1^*, \ldots, d_j^*\right)$ such that $\boldsymbol{x}^* \in$ zer $(\boldsymbol{A}+\boldsymbol{C})$ and $\boldsymbol{x}^k \xrightarrow{w} \boldsymbol{x}^*$.

Proof of Corollary 5.2.11

By making, for every j, $A = \partial g$, $C = \nabla \mu$, $B_j = \partial h_j$, and $E_j = \partial l_j$, it is clear that Algorithm 21 is an instance of Algorithm 20. The current corollary is proven by using the same arguments as in [122, Theorem 4.2].

Proof of Corollary 5.2.13

Define, for every $k \in \mathbb{N}$,

$$\begin{cases} \mathbf{y}^k = 2\mathbf{d}^{k+1} - \mathbf{d}^k, \\ \bar{\mathbf{d}}^k = \gamma \mathbf{d}^k, \\ \bar{\mathbf{y}}^k = \gamma \mathbf{y}^k, \end{cases} \tag{5.106}$$

and note that Lines 6 and 7 of Algorithm 22 can be rewritten as

$$\begin{aligned} \mathbf{d}^{k+1} &= \mathbf{d}^k + \mathbf{x}^{k+1} - \operatorname{prox}_{\frac{h}{\gamma}} \left(\mathbf{x}^{k+1} + \mathbf{d}^k\right) \\ &= \frac{1}{\gamma} \operatorname{prox}_{\gamma h^*} \left(\gamma \mathbf{x}^{k+1} + \bar{\mathbf{d}}^k\right). \end{aligned} \tag{5.107}$$

By following a similar reasoning to the one followed in Eqs. (5.50-5.55), we can then rewrite Lines 4-7 of Algorithm 22 as

$$\mathbf{v}^{k+1} = \operatorname{prox}_{\frac{h}{\gamma}} \left(\mathbf{x}^k + \mathbf{d}^k\right) \tag{5.108}$$

$$\mathbf{d}^{k+1} = \frac{1}{\gamma} \operatorname{prox}_{\gamma h^*} \left(\gamma \mathbf{x}^k + \bar{\mathbf{d}}^k\right) \tag{5.109}$$

$$\mathbf{p}^k = \operatorname{prox}_{\tau g} \left(\mathbf{x}^k - \tau \left(\nabla f(\mathbf{x}^k) + \gamma \left(\mathbf{x}^k - \mathbf{v}^{k+1} + \mathbf{d}^{k+1}\right)\right)\right) \tag{5.110}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Lambda^k \left(\mathbf{p}^k - \mathbf{x}^k\right). \tag{5.111}$$

It follows from Eqs. (5.108) and (5.109) that

$$\begin{aligned}
\mathbf{x}^k - \mathbf{v}^{k+1} + \mathbf{d}^{k+1} &= \mathbf{x}^k - \operatorname{prox}_{\frac{h}{\gamma}}\left(\mathbf{x}^k + \mathbf{d}^k\right) + \frac{1}{\gamma}\operatorname{prox}_{\gamma h_j^*}\left(\gamma\mathbf{x}^k + \bar{\mathbf{d}}^k\right)\\
&= \mathbf{x}^k + \frac{1}{\gamma}\operatorname{prox}_{\gamma h_j^*}\left(\gamma\mathbf{x}^k + \bar{\mathbf{d}}^k\right) - \mathbf{x}^k - \mathbf{d}^k + \frac{1}{\gamma}\operatorname{prox}_{\gamma h_j^*}\left(\gamma\mathbf{x}^k + \bar{\mathbf{d}}^k\right)\\
&= \frac{2}{\gamma}\operatorname{prox}_{\gamma h_j^*}\left(\gamma\mathbf{x}^k + \bar{\mathbf{d}}^k\right) - \mathbf{d}^k\\
&= 2\mathbf{d}^{k+1} - \mathbf{d}^k \qquad (5.112)\\
&= \mathbf{y}^k. \qquad (5.113)
\end{aligned}$$

Rewriting Eq. (5.110) using Eq. (5.113) yields

$$\mathbf{p}^k = \operatorname{prox}_{\tau g}\left(\mathbf{x}^k - \tau\left(\gamma\mathbf{y}^k + \nabla f(\mathbf{x}^k)\right)\right). \qquad (5.114)$$

Using (5.106), we can rewrite Eq. (5.112) as

$$\bar{\mathbf{y}}^k = 2\bar{\mathbf{d}}^{k+1} - \bar{\mathbf{d}}^k \qquad (5.115)$$

and Eq. (5.109) as

$$\bar{\mathbf{d}}^{k+1} = \operatorname{prox}_{\gamma h^*}\left(\gamma\mathbf{x}^k + \bar{\mathbf{d}}^k\right). \qquad (5.116)$$

We are now able to rewrite Eqs. (5.108)-(5.111) as shown in Algorithm 26, which is an instance of Algorithm 21 if one makes $N = 1, \mathcal{X} = \mathbb{R}^n, \mathcal{V} = \mathbb{R}^n, L_1 = \mathbf{I}_n, r_1 = \mathbf{0}, z = \mathbf{0}$, and $\forall_k, U_1^k = \gamma\mathbf{I}_n, U^k = \tau\mathbf{I}_n, e_1^k = \mathbf{0}, b_1^k = \mathbf{0}, \Lambda_1^k = \mathbf{I}_n, c^k = \mathbf{0}, a^k = \mathbf{0}$, and if (5.26) is satisfied for $j = 1$.

Algorithm 26: Algorithm 22 is an instance of Algorithm 21.

```
while stopping criterion is not satisfied do
    $\bar{\mathbf{d}}^{k+1} = \operatorname{prox}_{\gamma h^*}\left(\bar{\mathbf{d}}^k + \gamma\mathbf{x}^k\right)$;
    $\bar{\mathbf{y}}^k = 2\bar{\mathbf{d}}^{k+1} - \bar{\mathbf{d}}^k$;
    $\mathbf{p}^k = \operatorname{prox}_{\tau g}\left(\mathbf{x}^k - \tau\left(\bar{\mathbf{y}}^k + \nabla f(\mathbf{x}^k)\right)\right)$;
    $\mathbf{x}^{k+1} = \mathbf{x}^k + \Lambda^k\left(\mathbf{p}^k - \mathbf{x}^k\right)$;
end
```

The current corollary is proven by invoking Corollary 5.2.11.

Proof of Proposition 5.3.1

If one makes, in Algorithm 20, and for every k and j, $\mathcal{X} = \mathbb{R}^n$, $\mathcal{V}_j = \mathbb{R}^{m_j}$, $A = \partial g$, $B_j = \frac{\partial h_j}{\omega_j}$, $C = \nabla f$,

$$E_j : d \rightarrow \begin{cases} \mathcal{V}_j, & \text{if } d = 0,\\ \emptyset, & \text{if } d \neq 0, \end{cases} \qquad (5.117)$$

$L_j = \mathbf{D}_j$, $\Lambda^k = \lambda^k \mathrm{Id}$, $a^k = 0$, $b_j^k = 0$, $c^k = 0$, $e_j^k = 0$, $z = 0$, and $r_j = 0$, it is clear that Algorithm 24 is an instance of Algorithm 20. The present proposition is proven by invoking Corollary 5.2.10 and using the same arguments as in [122, Theorem 4.2].

5.5. Conclusions

In this chapter, we defined and analyzed operator-weighted averaged operators, and showed how they can be used to construct a number of algorithms. These algorithms have very broad applications and seem to be particularly suitable to address problems with sparsity-inducing regularizers, as suggested by a simple experiment. Possible future directions to explore are the possibility of relaxing two of the impositions on Λ^k—namely, assumptions (5.5) and (5.10)— and the study of which problems are more suitable to be tackled by these methods. We also presented a formalization of the idea that ADMM, when used to solve ℓ^2+regularizer minimization problems, is very similar to a second-order primal–dual method. Additionally, we provided some experimental evidence of this idea.

Part III.

Publications

This section has been omitted in the present version.

List of Acronyms

2-D two-dimensional

3-D three-dimensional

ADMM alternating-direction method of multipliers

ALI Advanced Land Imager

AM method by Almeida and Figueiredo [89]

BCCB block-circulant-circulant-block

BGS block-Gauss-Seidel

BPT binary partition tree

BSNR blurred-signal-to-noise ratio

BT Brovey transform method

BTTB block-Toeplitz-Toeplitz-block

CC cross correlation

CG conjugate-gradient method

CM method by Condat [67]

CNMF coupled non-negative matrix factorization

CS component substitution

DC direct-current

DFT discrete Fourier transform

EIA endmember-induction algorithm

EM electromagnetic

EO-1 Earth Observing-1 Mission

ERGAS erreur relative globale adimensionnelle de synthèse

FE filter estimation

FFT fast Fourier Transform

FIHS fast intensity-hue-saturation fusion technique

GFPCA guided filter PCA

GS Gram-Schmidt spectral sharpening method

GSA Gram-Schmidt adaptive method

HPF high-pass filtering method

HSI-GDL-EIA HSI superresolution methodologies by global dictionary learning using EIA

HSI-LDL-EIA HSI superresolution methodologies by local dictionary learning using EIA

HSI hyperspectral image

HySure hyperspectral superresolution method

i.i.d. independent and identically distributed

IHS intensity-hue-saturation

ISNR improvement in SNR

KKT Karush-Kuhn-Tucker

MAP maximum-*a-posteriori*

MRA multiresolution analysis

MSE mean-squared error

MSI multispectral image

MTF-GLP-HPM MTF-GLP with high-pass modulation

MTF-GLP MTF-generalized Laplacian

MTF modulation-transfer function

PAN panchromatic image

PCA principal-component analysis

PD positive definite

Proposed-AD proposed method with adaptive strategy

PSD positive semidefinite

PSF point-spread function

RGB red-gree-blue

RMSE root-mean-squared error

S&M Starck-Murtagh low-pass filter

SALSA split augmented-Lagrangian shrinkage algorithm

SAM spectral angle mapper

SFIM smoothing-filter-based intensity-modulation method

SNR signal-to-noise ratio

SR superresolution

SSNM semismooth Newton method

SVD singular-value decomposition

SW sliding windows

TV total variation

UIQI universal image quality index

VCA vertex component analysis

VMPD variable-metric primal-dual method

VTV vector total variation

ZBS method by Zhang et al. [106]

Bibliography

[1] G. A. Licciardi, M. A. Veganzones, M. Simões, J. Bioucas-Dias, and J. Chanussot, "Super-resolution of hyperspectral images using local spectral unmixing," in *IEEE Workshop Hyperspectral Image Signal Proces.: Evolution Remote Sens.*, Lausanne, Switzerland, June 2014.

[2] M. A. Veganzones, M. Simões, G. Licciardi, J. Bioucas-Dias, and J. Chanussot, "Hyperspectral super-resolution of locally low rank images from complementary multisource data," in *IEEE Int. Conf. Image Processing*, Paris, France, Oct. 2014, pp. 703–707.

[3] M. Simões, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "Hyperspectral image superresolution: An edge-preserving convex formulation," in *IEEE Int. Conf. Image Processing*, Paris, France, Oct. 2014, pp. 4166–4170.

[4] G. Vivone, M. Simões, M. Dalla Mura, R. Restaino, J. Bioucas-Dias, G. Licciardi, and J. Chanussot, "Pansharpening based on semiblind deconvolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 1997–2010, Apr. 2015.

[5] M. Simões, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "A convex formulation for hyperspectral image superresolution via subspace-based regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3373–3388, June 2015.

[6] L. Loncan, L. B. Almeida, J. Bioucas-Dias, X. Briottet, J. Chanussot, N. Dobigeon, S. Fabre, W. Liao, G. Licciardi, M. Simões, J.-Y. Tourneret, M. A. Veganzones, G. Vivone, Q. Wei, and N. Yokoya, "Comparison of nine hyperspectral pansharpening methods," in *IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, July 2015.

[7] ——, "Hyperspectral pansharpening: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 27–46, Sept. 2015.

[8] M. A. Veganzones, M. Simões, G. Licciardi, N. Yokoya, J. Bioucas-Dias, and J. Chanussot, "Hyperspectral super-resolution of locally low rank images from complementary multisource data," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 274–288, Jan. 2016.

[9] M. Simões, L. B. Almeida, J. Bioucas-Dias, and J. Chanussot, "A framework for fast image deconvolution with incomplete observations," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5266–5280, Nov. 2016.

[10] L. Breiman, "Statistical modeling: The two cultures," *Stat. Sci.*, vol. 16, no. 3, pp. 199–231, 2001.

[11] H. Bauschke and P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.* New York, NY, USA: Springer, 2011.

[12] M. Ulbrich, "Optimization methods in Banach spaces," in *Optimization with PDE Constraints.* Dordrecht, Netherlands: Springer, 2009, pp. 97–156.

[13] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems*, 2nd ed. Waltham, MA, USA: Academic Press, 2013.

[14] P. Hansen, J. Nagy, and D. O'Leary, *Deblurring Images: Matrices, Spectra and Filtering.* Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2006.

[15] M. Figueiredo and J. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3133–45, dec 2010.

[16] D. Lorenz and N. Worliczek, "Necessary conditions for variational regularization schemes," *Inverse Probl.*, vol. 29, no. 7, p. 075016, 2013.

[17] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.

[18] E. J. Candes and Y. Plan, "A probabilistic and RIPless theory of compressed sensing," *EEE Trans. Inf. Theory*, vol. 57, no. 11, pp. 7235–7254, Nov 2011.

[19] E. J. Candès, "Mathematics of sparsity (and a few other things)," in *Proc. Int. Congr. Math.*, Seoul, South Korea, 2014, pp. 2853–2856.

[20] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[21] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, March 2011.

[22] J. Kovačević and A. Chebira, "An introduction to frames," *Found. Trends Signal Process.*, vol. 2, no. 1, pp. 1–94, 2007.

[23] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.

[24] L. Guanter, H. Kaufmann, K. Segl, S. Foerster, C. Rogass, S. Chabrillat, T. Kuester, A. Hollstein, G. Rossner, C. Chlebek, C. Straif, S. Fischer, S. Schrader, T. Storch, U. Heiden, A. Mueller, M. Bachmann, H. Mühle, R. Müller, M. Habermeyer, A. Ohndorf, J. Hill, H. Buddenbaum, P. Hostert, S. van der Linden,

P. J. Leitão, A. Rabe, R. Doerffer, H. Krasemann, H. Xi, W. Mauser, T. Hank, M. Locherer, M. Rast, K. Staenz, and B. Sang, "The EnMAP spaceborne imaging spectroscopy mission for earth observation," *Remote Sensing*, vol. 7, no. 7, pp. 8830–8857, 2015.

[25] P. Stefano, P. Angelo, P. Simone, R. Filomena, S. Federico, S. Tiziana, A. Umberto, C. Vincenzo, N. Acito, D. Marco, M. Stefania, C. Giovanni, C. Raffaele, D. B. Roberto, L. Giovanni, and A. Cristina, "The PRISMA hyperspectral mission: Science activities and opportunities for agriculture and land monitoring," in *IEEE Int. Geosci. Remote Sens. Symp.*, July 2013, pp. 4558–4561.

[26] R. O. Green, G. Asner, S. Ungar, and R. Knox, "NASA mission to measure global plant physiology and functional types," in *IEEE Aerospace Conf.*, March 2008, pp. 1–7.

[27] A. Iwasaki, N. Ohgi, J. Tanii, T. Kawashima, and H. Inada, "Hyperspectral Imager Suite (HISUI) - Japanese hyper-multi spectral radiometer," in *IEEE Int. Geosci. Remote Sens. Symp.*, July 2011, pp. 1025–1028.

[28] G. A. Shaw and H. K. Burke, "Spectral imaging for remote sensing," *Lincoln Lab. J.*, vol. 14, no. 1, pp. 3–28, 2003.

[29] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, vol. 5, no. 2, pp. 354–379, 2012.

[30] N. Dobigeon, J.-Y. Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero, "Nonlinear unmixing of hyperspectral images," *IEEE Signal Process. Mag.*, no. January, pp. 82–94, 2011.

[31] J. Nascimento and J. Bioucas-Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, 2005.

[32] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Inverse problems and the use of a priori information," in *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. New York, NY, USA: Cambridge University Press, 2007, pp. 1001—-1006.

[33] F. Bach, "Sparse modeling for image and vision processing," *Found. Trends Comput. Graph. Vision*, vol. 8, no. 2—3, pp. 85–283, 2014.

[34] P. Campisi and K. Egiazarian, *Blind image deconvolution: Theory and applications.* Boca Raton, FL, USA: CRC Press, 2007.

[35] A. Rajagopalan and R. Chellappa, *Motion Deblurring: Algorithms and Systems.* Cambridge, UK: Cambridge University Press, 2014.

[36] P. Milanfar, *Super-Resolution Imaging.* Boca Raton, FL, USA: CRC Press, 2010.

[37] J. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, 2013.

[38] R. Rockafellar, *Convex Analysis.* New Jersey, USA: Princeton University Press, 1970.

[39] J. Nocedal and S. Wright, *Numerical Optimization.* New York, NY, USA: Springer, 2006.

[40] Y. Nesterov, *Introductory lectures on convex optimization: A basic course.* New York, NY, USA: Springer US, 2004, vol. 87.

[41] A. Nemirovski, "Optimization II: Standard numerical methods for nonlinear continuous optimization," *Technion – Israel Institute of Technology*, 1999. [Online]. Available: http://www2.isye.gatech.edu/~nemirovs/Lect_OptII.pdf

[42] J. E. Dennis and J. J. Moré, "A characterization of superlinear convergence and its application to quasi-Newton methods," *Math. Comp.*, vol. 28, no. 126, pp. 549–560, 1974.

[43] N. Komodakis and J. C. Pesquet, "Playing with duality: An overview of recent primal—-dual approaches for solving large-scale optimization problems," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 31—-54, Nov 2015.

[44] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds. New York, NY, USA: Springer New York, 2011, pp. 185–212.

[45] D. Donoho and I. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Am. Statist. Assoc.*, vol. 90, no. 432, pp. 1200–1224, 1995.

[46] M. Kowalski, "Sparse regression using mixed norms," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 303 – 324, 2009.

[47] R. D. Nowak and M. A. T. Figueiredo, "Fast wavelet-based image deconvolution using the EM algorithm," in *Proc. 35th Asilomar Conf. Signals, Systems, Computers, 2001*, vol. 1, Nov 2001, pp. 371–375.

[48] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug 2003.

[49] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[50] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM J. Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.

[51] P. L. Combettes, "Fejér monotonicity in convex optimization," in *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos, Eds. Boston, MA, USA: Springer US, 2009, pp. 1016–1024.

[52] E. K. Ryu and S. Boyd, "Primer on monotone operator methods," *Appl. Comput. Math.*, vol. 15, no. 1, p. 3–43, 2016.

[53] P. Giselsson and S. Boyd, "Linear convergence and metric selection for Douglas-Rachford splitting and ADMM," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 532–544, Feb 2017.

[54] P. Giselsson, M. Fält, and S. Boyd, "Line search for averaged operator iteration," in *IEEE 55th Conf. Decision Control (CDC)*, Dec 2016, pp. 1015–1022.

[55] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec 2007.

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

[57] D. A. Lorenz and T. Pock, "An inertial forward-backward algorithm for monotone inclusions," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 2, pp. 311–325, 2015.

[58] F. Alvarez, "Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in Hilbert space," *SIAM Journal on Optimization*, vol. 14, no. 3, pp. 773–782, 2004.

[59] P.-E. Maingé, "Convergence theorems for inertial KM-type algorithms," *J. Comput. Appl. Math.*, vol. 219, no. 1, pp. 223 – 236, 2008.

[60] R. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. Control Optimiz.*, vol. 14, no. 5, pp. 877–898, 1976.

[61] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 3, pp. 293–318, Jun. 1992.

[62] P. L. Combettes and J.-C. Pesquet, "A proximal decomposition method for solving convex variational inverse problems," *Inverse Problems*, vol. 24, no. 6, p. 065014, 2008.

[63] S. Setzer, G. Steidl, and T. Teuber, "Deblurring Poissonian images by split Bregman techniques," *Journal of Visual Communication and Image Representation*, vol. 21, no. 3, pp. 193 – 199, 2010.

[64] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems." *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–95, 2011.

[65] D. Gabay, "Applications of the method of multipliers to variational inequalities," in *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, ser. Studies in Mathematics and Its Applications, M. Fortin and R. Glowinski, Eds. Elsevier, 1983, vol. 15, pp. 299 – 331.

[66] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *SIAM J. Sci. Comput.*, vol. 66, no. 3, pp. 889–916, 2016.

[67] L. Condat, "A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optimiz. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013.

[68] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Adv. Comput. Math.*, vol. 38, no. 3, pp. 667–681, 2011.

[69] M. Schmidt, D. Kim, and S. Sra, "Projected newton-type methods in machine learning," in *Optimization for Machine Learning.* MIT Press, 2011, pp. 305––330.

[70] J. D. Lee, Y. Sun, and M. A. Saunders, "Proximal Newton-type methods for minimizing composite functions," *SIAM J. Optim.*, vol. 24, no. 3, pp. 1420–1443, 2014.

[71] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Image Process.*, vol. 57, no. 7, pp. 2479–2493, July 2009.

[72] P. Giselsson and S. Boyd, "Metric selection in fast dual forward–backward splitting," *Automatica*, vol. 62, pp. 1 – 10, 2015.

[73] M. Hintermüller, K. Ito, and K. Kunisch, "The primal-dual active set strategy as a semismooth Newton method," *SIAM J. Optimiz.*, vol. 13, no. 3, pp. 865–888, 2003.

[74] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," in *arXiv*, 2016. [Online]. Available: https://arxiv.org/abs/1606.04838

[75] R. Mifflin, "Semismooth and semiconvex functions in constrained optimization," *SIAM J. Control Optimiz.*, vol. 15, no. 6, pp. 959–972, 1977.

[76] M. Ulbrich, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces.* Philadelphia, PA, USA: MOS-SIAM Series on Optimization, 2011.

[77] F. H. Clarke, “Generalized gradients and applications,” *Trans. Am. Math. Soc.*, vol. 205, pp. 247–262, 1975.

[78] X. Chen, Z. Nashed, and L. Qi, “Smoothing methods and semismooth methods for nondifferentiable operator equations,” *SIAM J. Numer. Anal.*, vol. 38, no. 4, pp. 1200–1216, 2000.

[79] R. Griesse and D. Lorenz, “A semismooth Newton method for Tikhonov functionals with sparsity constraints,” *Inverse Probl.*, vol. 24, no. 3, p. 035007, 2008.

[80] C. Byrne, “A unified treatment of some iterative algorithms in signal processing and image reconstruction,” *Inverse Probl.*, vol. 20, no. 1, pp. 103——120, 2004.

[81] P. L. Combettes and B. C. Vũ, “Variable metric forward–backward splitting with applications to monotone inclusions in duality,” *Optimiz.*, vol. 63, no. 9, pp. 1289–1318, 2014.

[82] M. Burger, A. Sawatzky, and G. Steidl, “First order algorithms in variational image processing,” in *arXiv*, 2014. [Online]. Available: https://arxiv.org/abs/1412.4237

[83] A. Chambolle and T. Pock, “An introduction to continuous optimization for imaging,” *Acta Numer.*, vol. 25, pp. 161—-319, 005 2016.

[84] P. L. Combettes, D. Dũng, and B. C. Vũ, “Dualization of signal recovery problems,” *Set-Valued Var. Anal.*, vol. 18, no. 3, pp. 373–404, 2010.

[85] N. Parikh and S. Boyd, “Proximal algorithms,” *Found. Trends Optimiz.*, vol. 1, no. 3, pp. 123–231, 2013.

[86] M. Hintermüller, “Semismooth Newton methods and applications,” Department of Mathematics, Humboldt-University of Berlin, Tech. Rep. November, 2010. [Online]. Available: http://www.math.uni-hamburg.de/home/hinze/Psfiles/Hintermueller_OWNotes.pdf

[87] M. Ng, R. Chan, and W. Tang, “A fast algorithm for deblurring models with Neumann boundary conditions,” *SIAM J. Sci. Comput.*, vol. 21, no. 3, pp. 851–866, 1999.

[88] M. Donatelli and S. Serra-Capizzano, “On the treatment of boundary artifacts in image restoration by reflection and/or anti-reflection,” in *Matrix Methods: Theory, Algorithms and Applications*, V. Olshevsky and E. Tyrtyshnikov, Eds. Singapore: World Scientific Publishing Company, 2010.

[89] M. Almeida and M. Figueiredo, “Deconvolving images with unknown boundaries using the alternating direction method of multipliers.” *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3074–86, Aug. 2013.

[90] A. Matakos, S. Ramani, and J. Fessler, "Accelerated edge-preserving image restoration without boundary artifacts," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 2019–2029, 2013.

[91] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[92] S. Reeves, "Fast image restoration without boundary artifacts." *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1448–1453, 2005.

[93] A. Danielyan, V. Katkovnik, and K. Egiazarian, "BM3D frames and variational image deblurring," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1715–1728, April 2012.

[94] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *Proc. SPIE Electronic Imaging*, no. 6812-07, San Jose, CA, USA, 2008.

[95] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York, NY, USA: Cambridge University Press, 2004.

[96] L. Condat, "A generic proximal algorithm for convex optimization — application to total variation minimization," *Signal Processing Letters, IEEE*, vol. 21, no. 8, pp. 985–989, Aug 2014.

[97] Q. Wei, N. Dobigeon, and J. Y. Tourneret, "Fast fusion of multi-band images based on solving a Sylvester equation," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4109–4121, Nov. 2015.

[98] Q. Wei, N. Dobigeon, J. Y. Tourneret, J. Bioucas-Dias, and S. Godsill, "R-FUSE: Robust fast fusion of multiband images based on solving a Sylvester equation," *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1632–1636, Nov. 2016.

[99] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, March 2017.

[100] I. Amro, J. Mateos, M. Vega, R. Molina, and A. Katsaggelos, "A survey of classical methods and new trends in pansharpening of multispectral images," *EURASIP J. Adv. Signal Process.*, vol. 2011, no. 1, pp. 79–100, 2011.

[101] G. Vivone, L. Alparone, J. Chanussot, M. D. Mura, A. Garzelli, G. A. Licciardi, R. Restaino, and L. Wald, "A critical comparison among pansharpening algorithms," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2565–2586, May 2015.

[102] X. Bresson and T. Chan, "Fast dual minimization of the vectorial total variation norm and applications to color image processing," *Inverse Probl. and Imag.*, vol. 2, no. 4, pp. 455–484, 2008.

[103] J. Bioucas-Dias and J. Nascimento, "Hyperspectral subspace identification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, 2008.

[104] X. Zhao, F. Wang, T. Huang, M. Ng, and R. Plemmons, "Deblurring and sparse unmixing for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, 2013.

[105] N. Yokoya, T. Yairi, and A. Iwasaki, "Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 2, pp. 528–537, 2012.

[106] Y. Zhang, S. De Backer, and P. Scheunders, "Noise-resistant wavelet-based Bayesian fusion of multispectral and hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3834–3843, Nov. 2009.

[107] Y. Zhang, A. Duijster, and P. Scheunders, "A Bayesian restoration approach for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 9, pp. 3453–3462, Sep. 2012.

[108] Q. Wei, N. Dobigeon, and J. Tourneret, "Bayesian fusion of multi-band images," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 6, pp. 1117–1127, Sept 2015.

[109] T. Wang, G. Yan, H. Ren, and X. Mu, "Improved methods for spectral calibration of on-orbit imaging spectrometers," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 3924–3931, 2010.

[110] E. Middleton, S. Ungar, D. Mandl, L. Ong, S. Frye, P. Campbell, D. Landis, J. Young, and N. Pollack, "The Earth Observing One (EO-1) satellite mission: over a decade in space," *IEEE J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, vol. 6, no. 2, pp. 243–256, 2013.

[111] Q. Wei, J. Bioucas-Dias, N. Dobigeon, and J. Y. Tourneret, "Hyperspectral and multispectral image fusion based on a sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3658–3668, July 2015.

[112] Q. Wei, J. Bioucas-Dias, N. Dobigeon, J. Y. Tourneret, M. Chen, and S. Godsill, "Multiband image fusion based on spectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7236–7249, Dec. 2016.

[113] P. Addesso, M. Dalla Mura, L. Condat, R. Restaino, G. Vivone, D. Picone, and J. Chanussot, "Hyperspectral pansharpening using convex optimization and collaborative total variation regularization," in *IEEE Workshop Hyperspectral Image Signal Proces.: Evolution Remote Sens.*, Los Angeles, CA, USA, Aug. 2016.

[114] C. Lanaras, E. Baltsavias, and K. Schindler, "Hyperspectral super-resolution by coupled spectral unmixing," in *IEEE Int. Conf. Computer Vision*, Dec. 2015, pp. 3586–3594.

[115] H. Shen, X. Meng, and L. Zhang, "An integrated framework for the spatio-temporal-spectral fusion of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7135–7148, Dec. 2016.

[116] R. Guerra, S. López, and R. Sarmiento, "A computationally efficient algorithm for fusing multispectral and hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 5712–5728, Oct. 2016.

[117] Z. H. Nezhad, A. Karami, R. Heylen, and P. Scheunders, "Fusion of hyperspectral and multispectral images using spectral unmixing and sparse coding," *IEEE J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, vol. 9, no. 6, pp. 2377–2389, June 2016.

[118] Q. Wang, W. Shi, P. M. Atkinson, and Q. Wei, "Approximate area-to-point regression kriging for fast hyperspectral image sharpening," *IEEE J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, vol. 10, no. 1, pp. 286–295, Jan. 2017.

[119] K. Zhang, M. Wang, and S. Yang, "Multispectral and hyperspectral image fusion based on group spectral embedding and low-rank factorization," *IEEE Trans. Geosci. Remote Sens.*, 2016.

[120] C. Zou and Y. Xia, "Poissonian hyperspectral image superresolution using alternating direction optimization," *IEEE J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, vol. 9, no. 9, pp. 4464–4479, Sep. 2016.

[121] N. Yokoya, C. Grohnfeldt, and J. Chanussot, "Hyperspectral and multispectral data fusion: A comparative review of the recent literature," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 2, pp. 29–56, June 2017.

[122] P. L. Combettes and J.-C. Pesquet, "Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators," *Set-Valued Var. Anal.*, vol. 20, no. 2, pp. 307–330, 2011.

[123] P. L. Combettes and B. C. Vũ, "Variable metric quasi-Fejér monotonicity," *Nonlinear Anal-Theor*, vol. 78, pp. 17–31, 2013.

[124] O. Axelsson, *Iterative Solution Methods.* Cambridge University Press, 1994.

[125] V. S. Ryaben'kii and S. V. Tsynkov, *A Theoretical Introduction to Numerical Analysis.* Boca Raton, FL, USA: CRC Press, 2006.

[126] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, "A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation," *SIAM J. Sci. Comput.*, vol. 32, pp. 1832–1857, 2010.

[127] R. H. Byrd, G. M. Chin, J. Nocedal, and F. Oztoprak, "A family of second-order methods for convex ℓ_1-regularized optimization," *Math. Program.*, vol. 159, no. 1-2, pp. 435–467, Sep. 2016.

[128] N. Keskar, J. Nocedal, F. Öztoprak, and A. Wächter, "A second-order method for convex ℓ_1-regularized optimization with active-set prediction," *Optim. Methods Softw.*, vol. 31, no. 3, pp. 605–621, 2016.

[129] T. Chen, F. E. Curtis, and D. P. Robinson, "A reduced-space algorithm for minimizing ℓ_1-regularized convex functions," in *arXiv*, Feb. 2016. [Online]. Available: https://arxiv.org/abs/1602.07018

[130] R. W. R. Rockafellar, *Variational Analysis.* Berlin, Germany: Springer, 2009.

[131] L. Stella, A. Themelis, and P. Patrinos, "Forward–backward quasi-Newton methods for nonsmooth optimization problems," *Comput. Optimiz. Appl.*, pp. 1–45, 2017.

[132] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[133] H.-J. M. Shi, S. Tu, Y. Xu, and W. Yin, "A Primer on Coordinate Descent Algorithms," in *arXiv*, Sep. 2016. [Online]. Available: https://arxiv.org/abs/1610.00040

[134] P. L. Combettes and J.-C. Pesquet, "Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping," *SIAM J Optimiz.*, vol. 25, no. 2, pp. 1221–1248, 2015.

[135] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Nearly optimal sparse Fourier transform," in *Proc. 44th Annual ACM Symp. Th. Comput.*, ser. STOC '12. New York, NY, USA: ACM, 2012, pp. 563–578.

[136] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *ArXiv*, 2016. [Online]. Available: https://arxiv.org/abs/1611.02862

[137] P. Q. Muoi, D. N. Hao, P. Maass, and M. Pidcock, "Semismooth Newton and quasi-Newton methods in weighted ℓ_1-regularization," *J. Inverse Ill-Posed Probl.*, vol. 21, no. 5, pp. 665–693, 2013.

[138] P. Patrinos, L. Stella, and A. Bemporad, "Forward-backward truncated Newton methods for convex composite optimization," *ArXiv e-prints*, Feb. 2014. [Online]. Available: http://arxiv.org/abs/1402.6655

[139] E. Hans and T. Raasch, "Global convergence of damped semismooth Newton methods for ℓ_1 Tikhonov regularization," *Inverse Probl.*, vol. 31, no. 2, p. 025005, 2015.

[140] X. Xiao, Y. Li, Z. Wen, and L. Zhang, "A regularized semi-smooth Newton method with projection steps for composite convex programs," in *arXiv*, Mar. 2016. [Online]. Available: https://arxiv.org/abs/1603.07870

[141] M. Figueiredo, "Teaching a new trick to an old dog: Revisiting the quadratic programming formulation of sparse recovery using ADMM," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, May 2014, pp. 1512–1516.

[142] S. Wang and L. Liao, "Decomposition method with a variable parameter for a class of monotone variational inequality problems," *J. Optimiz. Theory App.*, vol. 109, no. 2, pp. 415–429, 2001.