Methods for ℓ_p/TV_p Regularized Optimization and Their Applications in Sparse Signal Processing

by

Jie Yan
B.Eng., Southeast University, China, 2008
M.A.Sc., University of Victoria, Canada, 2010

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Methods for $\ell_p/\mathbf{TV}_p$ Regularized Optimization and Their Applications in Sparse Signal Processing

by

Jie Yan
B.Eng., Southeast University, China, 2008
M.A.Sc., University of Victoria, Canada, 2010

Supervisory Committee

Dr. Wu-Sheng Lu, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Michael D. Adams, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. Yang Shi, Outside Member
(Department of Mechanical Engineering)

Supervisory Committee

Dr. Wu-Sheng Lu, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Michael D. Adams, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. Yang Shi, Outside Member
(Department of Mechanical Engineering)

ABSTRACT

Exploiting signal sparsity has recently received considerable attention in a variety of areas including signal and image processing, compressive sensing, machine learning and so on. Many of these applications involve optimization models that are regularized by certain sparsity-promoting metrics. Two most popular regularizers are based on the ℓ_1 norm that approximates sparsity of vectorized signals and the total variation (TV) norm that serves as a measure of gradient sparsity of an image.

Nevertheless, the ℓ_1 and TV terms are merely two representative measures of sparsity. To explore the matter of sparsity further, in this thesis we investigate relaxations of the regularizers to nonconvex terms such as ℓ_p and TV_p "norms" with $0 \leq p < 1$. The contributions of the thesis are two-fold. First, several methods to approach globally optimal solutions of related nonconvex problems for improved signal/image reconstruction quality have been proposed. Most algorithms studied in the thesis fall into the category of iterative reweighting schemes for which nonconvex problems are reduced to a series of convex sub-problems. In this regard, the second main contribution of this thesis has to do with complexity improvement of the ℓ_1/TV-regularized methodology for which accelerated algorithms are developed. Along with these investigations, new techniques are proposed to address practical implementation issues. These include the development of an ℓ_p-related solver that is easily parallelizable, and a matrix-based analysis that facilitates implementation for TV-related optimizations. Computer simulations are presented to demonstrate merits of the proposed models and algorithms as well as their applications for solving general linear inverse problems in the area of signal and image denoising, signal sparse representation, compressive sensing, and compressive imaging.

Contents

List of Abbreviations

BFGS	**B**royden-**F**letcher-**G**oldfarb-**S**hanno
BPDN	**B**asis **P**ursuit **D**e**N**oising
BP	**B**asis **P**ursuit
CI	**C**ompressive **I**maging
CS	**C**ompressive **S**ensing
DCT	**D**iscrete **C**osine **T**ransform
DMD	**D**igital **M**icromirror **D**evice
FGP	**F**ast **G**radient **P**rojection
FISTA	**F**ast **I**terative **S**hrinkage-**T**hresholding **A**lgorithm
GTV	**G**eneralized **T**otal **V**ariation
HDTV	**H**igher **D**egree **T**otal **V**ariation
inf	**inf**imum
IRL1	**I**teratively **R**eweighted ℓ_1-minimization
IRTV	**I**teratively **R**eweighted **T**otal **V**ariation
ISTA	**I**terative **S**hrinkage-**T**hresholding **A**lgorithm
i.i.d.	**i**ndependently and **i**dentically **d**istributed
LASSO	**L**east **A**bsolute **S**hrinkage and **S**election **O**perator
LB	**L**inearized **B**regman

LP	**L**inear **P**rogramming
L-BFGS	**L**imited memory **BFGS**
MFISTA	**M**onotone **FISTA**
MRI	**M**agnetic **R**esonance **I**maging
MSE	**M**ean **S**quare **E**rror
NCP	**N**onsmooth **C**onvex **P**rogramming
NoI	**N**umber **o**f **I**terations
PDE	**P**artial **D**ifferential **E**quation
PSNR	**P**eak **S**ignal-to-**N**oise **R**atio
P-P	**P**roximal-**P**oint
RIP	**R**estricted **I**sometry **P**roperty
ROF	**R**udin, **O**sher and **F**atemi
SB	**S**plit **B**regman
SNR	**S**ignal-to-**N**oise **R**atio
SOCP	**S**econd **O**rder **C**one **P**rogramming
TV	**T**otal **V**ariation
WTV	**W**eighted **T**otal **V**ariation

List of Tables

List of Algorithms

List of Figures

ACKNOWLEDGEMENTS

First and foremost, I am deeply indebted to my supervisor Dr. Wu-Sheng Lu for his continuous encouragement and tremendous support throughout the duration of my Ph.D. journey. His sharp insight on my thesis topic, thoughtfulness in conducting research, and dedicated guidance towards high-standard work have made this thesis a reality. He trusted me with maximum flexibility to pursue my interests and provided invaluable instructions to help me grow professionally. I feel very fortunate working under his supervision.

I am very thankful to other members in my committee, Dr. Michael D. Adams for his colorful lectures that deepened my understanding in digital signal processing, Dr. Yang Shi for his insightful comments that helped me improve the content of this thesis, and Dr. Wei-Ping Zhu for agreeing to serve as my external examiner. I would also like to thank my undergraduate advisor Professor Chunming Zhao who provided me with support and motivation in pursuing a Ph.D. degree.

It is a pleasure to express my gratitude to many professors I was able to have at the University of Victoria, Dr. Xiaodai Dong, Dr. Panajotis Agathoklis, Dr. Dale Olesky, Dr. Alexandra Branzan Albu, and Dr. Andreas Antoniou, for sharing their expertise that helped me broaden my knowledge base and gain analytical perspective during my M.A.Sc. and Ph.D. studies.

I would also like to thank the staff of the Department of Electrical and Computer Engineering, Catherine, Moneca, Janice, Dan, Lynne, Vicky, Amy, Erik and Kevin for providing warm assistance and professional support during my graduate studies. I am lucky to meet new friends and colleagues in the last few years, Li, Ping, Weicong, Bojiang, Yihai, Ana-Maria, Ioana, Yimin, Binyan, Chenyuan, Niko, Fieran and Congzhi. Their friendship has made my life in Canada colorful.

Most importantly, I owe my deepest gratitude to my parents for inspiring me to follow my dreams and supporting me unconditionally in all of my pursuits. I am thankful beyond words to my wife, Lu Cao, for her patience, love and encouragement, from whom I have learned a lot and improved myself. I would not have been where I am today without your support.

Dedicated to
my beloved mother, father, and sister,
and to my beloved wife,
and my precious son.

Chapter 1

Introduction

In this thesis, we consider solving linear inverse problems using nonconvex ℓ_p and TV_p regularizers with $0 < p < 1$, and applications of the proposed optimization models in the general area of signal recovery including signal denoising and compressive sensing of 1-D signals and 2-D images. The purpose of this chapter is to introduce the literature relevant to the problems considered, discuss the motivations for improving the existing methods, and describe the main contributions and structure of the thesis.

1.1 The ℓ_1-ℓ_2 Problem and its Nonconvex Relaxation

Modeling signals by exploring sparsity through ℓ_1-norm has emerged as an effective framework in signal processing over the last several decades. The rationale of sparse modeling is that, in many instances, the signal we wish to recover is sparse by itself or sparse in a certain transformation domain. The ℓ_1 norm was adopted as early as in 1979 by Taylor, Banks and McCoy [104] to deconvolve seismic trace signals. In late 1980's, initial theoretical support was provided by Donoho and Stark in [45], and more rigorous analysis was refined in subsequent years [44, 54, 65, 107]. Advanced algorithms as the LASSO [105] and basis pursuit [39] for ℓ_1 minimization began to broaden in mid 1990's.

The use of ℓ_1-regularization is arguably considered the "modern least squares" [24] because of its wide applications, especially during the recent development in the field of compressive sensing (CS) [21, 22, 47]. In brief, CS reconstructs a signal from a relatively small number of linear measurements which appear to be highly incomplete compared to that dictated by the Shannon-Nyquist sampling theory. Most of the recovery problems have led to an ℓ_1-ℓ_2 formulation as

$$\underset{\mathbf{s}}{\text{minimize}} \;\; F(\mathbf{s}) = \lambda\|\mathbf{s}\|_1 + \|\mathbf{\Theta}\mathbf{s} - \mathbf{y}\|^2 \tag{1.1}$$

where $\mathbf{s}$ is a sparse representation of signal $\mathbf{x}$ under sparsifying transformation $\mathbf{\Psi}$, namely $\mathbf{x} = \mathbf{\Psi}\mathbf{s}$, and $\mathbf{y}$ denotes the measurement vector.

An attractive feature of the formulation in (1.1) is that $F(\mathbf{s})$ is convex and its global minimizer can be identified using a convex-program solver. Various classical iterative optimization algorithms exist for the ℓ_1-ℓ_2 sparse approximation problem, e.g., homotopy solvers and greedy techniques like matching pursuit and orthogonal matching pursuit [80]. Over the past several years, iterative-shrinkage algorithms have emerged as a family of highly effective numerical methods for ℓ_1-ℓ_2 problems, and are shown to be efficient and practical for large-scale image processing applications [124]. Of particular interest is a proximal-point-function based algorithm known as the fast iterative shrinkage-thresholding algorithm (FISTA) developed in [8,9], which is shown to provide a convergence rate of $O(1/k^2)$ where k denotes the number of iterations, compared to the rate of $O(1/k)$ by the well-known iterative shrinkage-thresholding algorithm (ISTA), while maintaining practically the same complexity as the ISTA. A more comprehensive discussion of the iterative-shrinkage algorithms will be provided in Chapter 2.

Let the sparsity of vector $\mathbf{s}$ be defined as the number of nonzero entries in $\mathbf{s}$ and denote it by K. Obviously the sparsity of $\mathbf{s}$ is connected to its "ℓ_0 norm" by $\|\mathbf{s}\|_0 = K$, and this explains why the ℓ_0 norm is inherently involved in many signal processing problems as long as sparsity plays a role. Nevertheless, it is well known that optimization problems with ℓ_0 regularizers are NP-hard [83]. In this regard, the convex relaxation of "ℓ_0 norm" to ℓ_1 norm is a natural way to convert an NP-hard problem to a convex problem of polynomial complexity. Through the work of Candès, Romberg, and Tao [21,23,27], the ℓ_1-norm based convex relaxation methodology has been theoretically justified and gained a great deal of attention as it finds wide range of applications.

Between the ℓ_1 norm and "ℓ_0 norm" there is wide range of "ℓ_p norm" with $0 < p < 1$. On one hand, the "ℓ_p norm" more accurately approximates the "ℓ_0 norm" as p gets smaller hence such an "ℓ_p norm" is expected to better promote sparsity. On the other hand, $\|\mathbf{s}\|_p$ is nonconvex as long as $p < 1$, hence the problem with such an ℓ_p regularizer is nonconvex and the optimization procedure for such problems becomes much more involved. It is with this motivation the nonconvex relaxation of the problem in (1.1), namely,

$$\underset{\mathbf{s}}{\text{minimize}}\ \ F(\mathbf{s}) = \lambda\|\mathbf{s}\|_p^p + \|\boldsymbol{\Theta}\mathbf{s} - \mathbf{y}\|^2 \tag{1.2}$$

has been investigated and improved performance relative to its ℓ_1-ℓ_2 counterpart is reported in [35, 36, 57, 91, 92, 114–116]. In this thesis we study the ℓ_p-ℓ_2 formulation with orthogonal bases and overcomplete dictionaries, respectively. With a variety of system settings we demonstrate that compared to classical ℓ_1-regularized optimization, finding satisfactory local minimizers for an ℓ_p-regularized problem enables us to exactly reconstruct sparse signals with fewer measurements and to denoise corrupted signals with improved signal-to-noise ratio (SNR).

1.2 The TV-regularized Problem and its Nonconvex Relaxation

The total variation (TV) model introduced by Rudin, Osher and Fatemi (ROF) [98] is a regularization approach for image processing in which the standard ℓ_2-norm fidelity is regularized by the TV of the image. This model has proven to be capable of properly preserving image edges and successful in a wide range of image recovery/reconstruction applications. The discrete model of TV regularization can be cast into an unconstrained optimization problem

$$\underset{\mathbf{U}}{\text{minimize}}\ \ \text{TV}(\mathbf{U}) + \frac{\mu}{2}\|\mathcal{A}(\mathbf{U}) - \mathbf{B}\|_{\text{F}}^2 \tag{1.3}$$

or in constrained formulation

$$\begin{aligned} &\underset{\mathbf{U}}{\text{minimize}} && \text{TV}(\mathbf{U}) && \text{(1.4a)} \\ &\text{subject to:} && \|\mathcal{A}(\mathbf{U}) - \mathbf{B}\|_{\text{F}}^2 < \sigma^2 && \text{(1.4b)} \end{aligned}$$

where $\mathcal{A}$ is a linear operator applied to image $\mathbf{U}$ and $\mathbf{B} \in R^{m\times n}$ corresponds to the observed image. The discretized anisotropic and isotropic TV of image $\mathbf{U}$ are defined as [9]

$$\mathrm{TV}^{(\mathrm{A})}(\mathbf{U}) = \sum_{i=1}^{m-1}\sum_{j=1}^{n} |U_{i,j} - U_{i+1,j}| + \sum_{i=1}^{m}\sum_{j=1}^{n-1} |U_{i,j} - U_{i,j+1}| \tag{1.5}$$

and

$$\begin{aligned} \mathrm{TV}^{(\mathrm{I})}(\mathbf{U}) = &\sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \sqrt{|U_{i,j} - U_{i+1,j}|^2 + |U_{i,j} - U_{i,j+1}|^2} \\ &+ \sum_{i=1}^{m-1} |U_{i,n} - U_{i+1,n}| + \sum_{j=1}^{n-1} |U_{m,j} - U_{m,j+1}| \end{aligned} \tag{1.6}$$

respectively.

The ROF model has received a great deal of attention for image denoising, image deblurring, and compressive imaging which allows images to be reconstructed from relatively few sampled data [21, 26, 47]. In this thesis, we consider the TV-based denoising problem for which $\mathcal{A}$ is simply the identity operator $\mathcal{I}$, and the compressive imaging problem where $\mathcal{A}$ corresponds to the sampling operation adopted in a magnetic resonance imaging (MRI) application.

Solving a TV-based regularization appears to be challenging because the TV norm is nonsmooth. Furthermore, it is inherently of large scale which renders the task of developing time and memory efficient methods nontrivial. Sustained research efforts have been made in developing first-order algorithms that require less memory but exhibit faster convergence for large-scale computation. Chambolle [28, 29] developed a gradient-based algorithm to solve the denoising problem and established faster convergence than primal-based schemes, see [30, 33, 67]. Beck and Teboulle [8] extended the dual-based approach of Chambolle to constrained optimization problems, that combines the acceleration mechanism FISTA with a fast gradient projection (FGP) method which demonstrates a faster rate of convergence than traditional gradient-based methods. Of particular interest is an algorithm named Split Bregman method developed by Goldstein and Osher [60]. The algorithm leverages the Bregman iteration scheme [19, 34, 89, 90, 120] for ℓ_1-regularized problems and can be extended to problems involving TV regularization term. The Split Bregman method has been recognized as one of the fastest solvers for problems considered herein.

Inspired by the ability of ℓ_p-regularized algorithms [35, 36, 57, 91, 92, 114–116] and the close connection of TV to the ℓ_1 norm, we extend the concept of conventional TV to a generalized TV (GTV) that involves pth power (with $p < 1$) of the discretized gradient of the image, and study the TV_p-regularized problems as

$$\underset{\mathbf{U}}{\text{minimize}}\ \ \mathrm{TV}_p(\mathbf{U}) + \frac{\mu}{2}\|\mathcal{A}(\mathbf{U}) - \mathbf{B}\|_{\mathrm{F}}^2 \tag{1.7}$$

or in constrained formulation

$$\begin{aligned} \underset{\mathbf{U}}{\text{minimize}} \quad & \mathrm{TV}_p(\mathbf{U}) && \text{(1.8a)} \\ \text{subject to:} \quad & \|\mathcal{A}(\mathbf{U}) - \mathbf{B}\|_{\mathrm{F}}^2 < \sigma^2 && \text{(1.8b)} \end{aligned}$$

The reader is referred to Chapter 5, Sec. 5.1 for definition of $\mathrm{TV}_p(\mathbf{U})$. Because the term $\mathrm{TV}_p(\mathbf{U})$ is nonconvex, the problems in (1.7) and (1.8) are generally difficult to tackle directly within existing TV-regularization framework. In the thesis, we propose a weighted TV (WTV) iterative strategy to locally approximate the TV_p-regularized problem, and demonstrate its ability to handle large-scale images. We present numerical examples to demonstrate improved performance for image denoising and image reconstruction of the new algorithms with $p < 1$ relative to that obtained by the standard TV minimization.

1.3 Contributions and Organization of the Thesis

1.3.1 Contributions of the Thesis

The work presented in the thesis is concerned with ℓ_p/TV_p regularized optimization with a focus on two aspects of the problems, namely, to improve signal reconstruction performance by finding nearly global minimizer of relaxed problems and to develop accelerated algorithms and demonstrate their efficiency for large-scale problems. In summary, the main contributions of the thesis include

- Development of a fast solver for global minimization of ℓ_p-ℓ_2 problem in case of an orthogonal basis;
- Design of a power-iterative strategy in conjunction with FISTA-type minimization framework to solve the ℓ_p-ℓ_2 problem and reach solution likely globally optimal in case of an overcomplete basis;
- Development of a smoothed ℓ_p-ℓ_2 solver which exhibit less oscillation in SNR profiles of denoised signals;

- Development of a dual-based linearized Bregman method to accelerate computation of signal reconstruction based on compressed samples, especially for large-scale signals;
- Proposal of the concept of generalized total variation (GTV), or pth-power TV, and development of an iteratively reweighting algorithm to approximate global solution of nonconvex GTV-regularized problem for image denoising;
- Development of a matrix-based analysis for the sparse MRI reconstruction problem and a weighted TV minimization framework using a Split Bregman type iteration to solve the nonconvex GTV minimization problem for compressive imaging.

1.3.2 Organization of the Thesis

The thesis is organized as follows

Chapter 2 - Preliminaries

In this chapter, background information and preliminary knowledge of direct relevance to the problems to be examined in the thesis are introduced. These include an iterative shrinkage-thresholding algorithm and an accelerated method, an optimization model of the ℓ_1-ℓ_2 problem and its applications, a framework of compressive sensing for signal/image reconstruction, the Bregman iteration and linearized Bregman algorithm for equality constrained nonsmooth convex programming, and total variation regularized optimization with applications to image denoising and compressive imaging.

Chapter 3 - Methods for ℓ_p-ℓ_2 Regularized Problems

The chapter investigates a nonconvex extension of the ℓ_1 norm to an ℓ_p regularization term with $0 \leq p < 1$. We first propose a fast solver for global solution of the ℓ_p-ℓ_2 problem where an orthogonal basis is considered. In the case of an overcomplete dictionary, we integrate the global solver into a FISTA-type iteration framework, and develop a power-iterative strategy to reach solutions that are likely globally optimal. Performance of the proposed techniques is evaluated for signal sparse representation and compressive sensing. The second part of this chapter is presented with a smoothed ℓ_p-ℓ_2 solver for signal denoising, using which oscillations in the ℓ_p SNR profiles by the conventional global solver are suppressed as much as possible. We simulate the algorithm on 1-D and 2-D signals and show its usefulness in signal denoising.

Chapter 4 - Fast Dual-Based Linearized Bregman Algorithms for Compressive Sensing

An equality constrained nonsmooth convex problem that is central to compressive sensing is examined in this chapter. We start with an analysis of its dual problem that is followed by discussing a dual-based linearized Bregman method. We then propose a fast algorithm to accelerate the conventional linearized Bregman iterations by introducing additional steps adopted in FISTA-type iterations. It is shown that the convergence rate is improved from $O(1/k)$ to $O(1/k^2)$ where k is the number of iteration. Experimental results are presented to support the proposed algorithm's efficiency in converging to globally optimal solution and its capability for large-scale compressive sensing.

Chapter 5 - Image Denoising by Generalized Total Variation Regularization

This chapter investigates a nonconvex extension of the TV-regularization problem for image denoising. First, we generalize the standard TV to a pth-power TV with $0 \leq p < 1$ that promotes sparser gradient information. Next, we propose to approximate solution of the nonconvex generalized TV (GTV)-regularized problem by solving iteratively reweighted TV (IRTV) convex subproblems. In particular, a power-iterative strategy is developed for the IRTV algorithm to converge to a reasonably good local solution if not the global solution, and a modified Split Bregman method is developed to properly handle the presence of nontrivial weights in weighted TV. Finally, we demonstrate improved performance compared to several well-known methods for image denoising.

Chapter 6 - Compressive Imaging by Generalized Total Variation Regularization

This chapter examines the sparse MRI reconstruction problem as an application of compressive sensing for images, also named as compressive imaging. We first present a matrix-based analysis of TV regularization model for which image variables are regarded as matrices rather than column-stacked vectors, and demonstrate its computational efficiency in terms of time and memory requirements. We then apply the GTV regularizer to a Fourier-based MRI reconstruction problem. The chapter concludes with experimental studies on reconstructing a variety of synthetic and natural images using the proposed method. Significant performance gain relative to existing algorithms is exhibited.

Chapter 7 - Conclusions

Finally, this chapter concludes the thesis and suggests several directions for future research.

Chapter 2

Preliminaries

In this chapter, we present preliminaries that provide background information for the problems to be studied in the subsequent chapters of the thesis. These include iterative and fast iterative shrinkage-thresholding algorithms, ℓ_1-ℓ_2 optimization problem and its applications, signal acquisition and recovery with compressive sensing, linearized Bregman algorithm, and total variation regularized problems for image denoising and compressive imaging.

2.1 The ℓ_1-ℓ_2 Optimization Problem

Over the last two decades, modeling signals exploring sparsity has emerged as an effective technique in signal processing. A central point in sparse signal processing is to identify an approximate solution to an ill-posed or under-determined linear system while requiring that the solution has fewest nonzeros entries. This problem arises in various areas across engineering and science [39,108]. Many applications in signal and image processing, such as denoising, inpainting, deblurring and compressive sensing, all lead to a mixed ℓ_1-ℓ_2 unconstrained convex problem as

$$\underset{\mathbf{s}}{\text{minimize}} \;\; F(\mathbf{s}) = \lambda||\mathbf{s}||_1 + ||\mathbf{\Theta s} - \mathbf{y}||^2 \tag{2.1}$$

where $\mathbf{s} \in R^N$, $\mathbf{\Theta} \in R^{M \times N}$ and $\mathbf{y} \in R^M$. Parameter $\lambda > 0$ in (2.1) is a regularization parameter that controls the tradeoff between the sparsity of $\mathbf{s}$ and the approximation error $||\mathbf{\Theta s} - \mathbf{y}||^2$. The ℓ_1 norm of vector $\mathbf{s}$ is defined as $\|\mathbf{s}\|_1 = \sum_{i=1}^N |s_i|$.

As a variant of the well-known basis pursuit (BP) problem [39], (2.1) is a nonsmooth (because $\|\mathbf{s}\|_1$ as a function of $\mathbf{s}$ is nondifferentiable), convex, unconstrained problem for which many efficient global solution techniques exist [124]. In principle, the ℓ_1-ℓ_2 problem can be solved using various classical iterative optimization algorithms [39], homotopy solvers [51, 53] and greedy techniques like matching pursuit and orthogonal matching pursuit [123]. However, these algorithms are often impractical in high-dimensional problems, as often encountered in image processing applications [124]. One of the state-of-the-art techniques in dealing with large-scale ℓ_1-ℓ_2 problems is the fast iterative-shrinkage-thresholding algorithm (FISTA) [8] which will be introduced in Sec. 2.4.

On the application front, several authors have successfully applied the ℓ_1-ℓ_2 model to a variety of problems encountered in signal and image processing, such as denoising, deblurring, compressive sensing, sparse representation, source-separation and more. Several applications of the ℓ_1-ℓ_2 model that are relevant to this thesis are described below.

Signal Denoising

Let $\mathbf{y}$ be the observation of a signal $\mathbf{x}$ that is contaminated by Gaussian white noise $\mathbf{w}$, i.e., $\mathbf{y} = \mathbf{x} + \mathbf{w}$. Without loss of generality, assume that $\mathbf{x}$ admits a sparse or nearly sparse representation in a suitable dictionary $\boldsymbol{\Psi}$, namely $\mathbf{x} = \boldsymbol{\Psi}\mathbf{s}$ where $\mathbf{s}$ is sparse. The well-known basis pursuit denoising (BPDN) [39] to recover signal $\mathbf{x}$ from noisy measurement $\mathbf{y}$ refers to the solution of

$$\underset{\mathbf{s}}{\text{minimize}} \;\; \lambda||\mathbf{s}||_1 + ||\boldsymbol{\Psi}\mathbf{s} - \mathbf{y}||^2$$

where parameter $\lambda > 0$ depends on the variance of noise $\mathbf{w}$ as well as the cardinality of dictionary $\boldsymbol{\Psi}$ [39]. As we can see, the objective function fits into the model (2.1) with $\boldsymbol{\Theta} = \boldsymbol{\Psi}$.

Compressive Sensing

As an alternative and effective data acquisition strategy, compressive sensing (CS) acquires a signal by collecting a relatively small number of linear measurements. The signal is later recovered with a nonlinear process [21, 47]. More specifically, rather than direct sampling with a Nyquist rate, compressive sensing suggests that we sense the vector $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x}$ where $\boldsymbol{\Phi} \in R^{M \times N}$ contains a set of $M \ll N$ projection directions onto which the signal is projected [22, 47]. In this way, compressive sensing facilitates us to sample a signal while compressing it. Reconstruction of the signal from its samples, i.e., $\mathbf{y}$, is achieved by solving

$$\begin{array}{ll} \text{minimize} & ||\mathbf{s}||_1 \\ \text{subject to:} & ||\boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{s} - \mathbf{y}||_2 \leq \varepsilon \end{array}$$

assuming that $\mathbf{x}$ can be sparsely represented under basis (or dictionary) $\boldsymbol{\Psi}$. Regardless of whether or not the measurements are noise-free, the recovery problem can be solved in the ℓ_1-ℓ_2 formulation (2.1) with $\boldsymbol{\Theta} = \boldsymbol{\Phi}\boldsymbol{\Psi}$.

The relationship between the ℓ_1-ℓ_2 model and signal recovery through compressive sensing will be elaborated further in Sec. 2.2.

Signal Sparse Representation

Another typical sparse representation problem is to find the sparsest representation of a discrete signal $\mathbf{x}$ under a (possibly overcomplete) dictionary $\boldsymbol{\Psi}$. The sparsity of a vector $\mathbf{s}$ refers to the number of nonzero entries in $\mathbf{s}$, which is often expressed as the ℓ_0 norm of $\mathbf{s}$ defined by $\|\mathbf{s}\|_0$, although strictly speaking the ℓ_0 norm is not a vector norm. With this notation, the problem considered here can be described as minimizing $\|\mathbf{s}\|_0$ subject to $\mathbf{x} = \boldsymbol{\Psi}\mathbf{s}$. Another version of the problem permits a small amount of perturbation in the measurements, i.e., $\mathbf{x} = \boldsymbol{\Psi}\mathbf{s} + \mathbf{w}$ and the problem becomes

$$\begin{array}{ll} \underset{\mathbf{s}}{\text{minimize}} & \|\mathbf{s}\|_0 \\ \text{subject to:} & \|\boldsymbol{\Psi}\mathbf{s} - \mathbf{x}\| \leq \varepsilon \end{array}$$

Unfortunately, both problems are nonconvex and known to be NP hard. This motivates the development of efficient algorithms for suboptimal solutions of the problem. An appealing solution method is the basis pursuit (BP) algorithm [39] which solves a modified version of the above problem with the ℓ_0 norm replaced by a convex ℓ_1 norm. The problem thus modified can be formulated as a quadratic convex problem, known as second order cone programming (SOCP) problem, which admits a unique global solution. In principle, the BP problem can be solved using a standard solver for convex problems. Recent studies exploring the specific structure of the problem have led to more efficient algorithms [80, 106]. Among these, the ℓ_1-ℓ_2 optimization is a popular approach that converts the constrained minimization into an unconstrained convex problem as

$$\underset{\mathbf{s}}{\text{minimize}} \;\; \lambda||\mathbf{s}||_1 + ||\mathbf{\Psi s} - \mathbf{x}||^2$$

which is the same as the ℓ_1-ℓ_2 model in (2.1).

In summary, the ℓ_1-ℓ_2 model in (2.1) is fundamental to these applications thus we are strongly motivated to develop new algorithms to deal more efficiently with this optimization problem.

2.2 Signal Acquisition and Recovery with Compressive Sensing

The foundation of current *compressive sensing* (CS) theory, also known as *compressive sampling* or *compressed sensing*, was laid by three papers [21], [47] and [22] in 2006 that, together with several other papers, have inspired a burst of intensive research activities in CS in the past several years [77].

The classical sampling method requires sampling a bandlimited signal at a rate (known as the Nyquist rate) greater than or equal to twice the bandwidth of the signal. Rather than evenly sampling at the Nyquist rate which can be prohibitively high for signals with broad spectrum, compressive sensing acquires a signal of interest indirectly by collecting a relatively small number of its projections. In particular, compressive sensing (CS) based signal acquisition computes M linear measurements of an unknown signal $\mathbf{x} \in R^N$ with $M < N$. This acquisition process can be described as

$$\mathbf{y} = \mathbf{\Phi x} \quad \text{with} \quad \mathbf{\Phi} = [\boldsymbol{\phi}_1 \; \boldsymbol{\phi}_2 \; \ldots \; \boldsymbol{\phi}_M]^T \tag{2.4}$$

where $\boldsymbol{\phi}_k \in R^N (k = 1, 2, \ldots, M)$. Suppose signal $\mathbf{x}$ is K-sparse with respect to an orthonormal basis $\{\boldsymbol{\psi}_j\}_{j=1}^N$ ($\boldsymbol{\psi}_j \in R^N$), then $\mathbf{x}$ can be expressed as

$$\mathbf{x} = \mathbf{\Psi s} \tag{2.5}$$

where $\mathbf{\Psi} = [\boldsymbol{\psi}_1 \; \boldsymbol{\psi}_2 \; \ldots \; \boldsymbol{\psi}_N]$ is an orthogonal matrix and $\mathbf{s}$ is a K-sparse signal with $K \ll N$ nonzero elements. The CS theory mandates that if matrix $\mathbf{\Theta} = \mathbf{\Phi\Psi}$ obeys the restricted isometry property (RIP) of order $2K$, i.e. the inequality

$$(1 - \delta_{2K})||\mathbf{s}||_2^2 \leq ||\mathbf{\Theta s}||_2^2 \leq (1 + \delta_{2K})||\mathbf{s}||_2^2$$

holds for all $2K$-sparse vectors $\mathbf{x}$ with $\delta_{2K} < \sqrt{2} - 1$, then $\mathbf{s}$ can be exactly recovered via the convex optimization

$$\begin{aligned} \text{minimize} \quad & ||\mathbf{s}||_1 && \text{(2.6a)} \\ \text{subject to:} \quad & \mathbf{\Theta s} = \mathbf{y} && \text{(2.6b)} \end{aligned}$$

and $\mathbf{x}$ is recovered by Eq. (2.5).

A sensing matrix $\mathbf{\Phi}$ obeys RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$ if it is constructed by (i) sampling i.i.d. entries from the normal distribution with zero mean and variance $1/M$, or (ii) sampling i.i.d. entries from a symmetric Bernoulli distribution (i.e. $\text{Prob}(\phi_{ij} = \pm 1/\sqrt{M}) = 1/2$), or (iii) sampling i.i.d. from other sub-Gaussian distribution, or (iv) sampling a random projection matrix $\mathbf{P}$ that is incoherent with matrix $\mathbf{\Psi}$ and normalizing it as $\mathbf{\Phi} = \sqrt{N/M}\mathbf{P}$, with $M \geq CK \log(N/K)$ and C a constant [23].

In practice, $\mathbf{x}$ is likely only approximately K-sparse under $\mathbf{\Psi}$. In addition, measurement noise may be introduced in the sensing process as $\mathbf{y} = \mathbf{\Phi x} + \mathbf{w}$. In this case the procedure of reconstructing $\mathbf{s}$ is performed by solving convex problem

$$\begin{aligned} \text{minimize} \quad & ||\mathbf{s}||_1 && \text{(2.7a)} \\ \text{subject to:} \quad & ||\mathbf{\Theta s} - \mathbf{y}||_2 \leq \varepsilon && \text{(2.7b)} \end{aligned}$$

where ε stands for the permissible deviation. This problem was first discussed in [39] as basis pursuit (BP). A variant of problem (2.7) mixes ℓ_1 and ℓ_2 expressions in the form of (2.1) where the constraint is replaced with a penalty term. The parameter λ replaces the threshold ε in (2.7) which controls the tradeoff between the reconstruction error and signal sparsity.

2.3 Iterative Shrinkage-Thresholding Algorithm

Over the past several years, a family of iterative-shrinkage algorithms have emerged as highly effective numerical methods for the ℓ_1-ℓ_2 problem. We begin with reviewing an algorithm, known as the iterative shrinkage-thresholding algorithm (ISTA), which also bears the names of “proximal-point method” and “separable surrogate functionals method" [124]. Consider the general formulation

$$\underset{\mathbf{x}\in R^n}{\text{minimize}} \;\; F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}) \tag{2.8}$$

and make the following assumptions on functions $f(\cdot)$ and $g(\cdot)$:

- $f(\cdot) : R^n \to R$ is a smooth convex function and is continuously differentiable with Lipschitz continuous gradient, i.e., there exist a constant L such that

  $$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$$

  for every $\mathbf{x}, \mathbf{y} \in R^n$ where $\|\cdot\|$ denotes the standard Euclidean norm and $L > 0$ is called the Lipschitz constant for gradient $\nabla f(\mathbf{x})$.

- $g(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is a continuous convex function which is possibly nonsmooth.

Consider the following quadratic approximation of $F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$ at a given point $\mathbf{y}$:

$$Q_L(\mathbf{x}, \mathbf{y}) = f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y})\rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2 + g(\mathbf{x})$$

which is convex quadratic, hence admits a unique minimizer as $p_L(\mathbf{y}) = \text{argmin}\, Q_L(\mathbf{x}, \mathbf{y})$. The unique minimizer $p_L(\mathbf{y})$ can be equivalently cast as

$$p_L(\mathbf{y}) = \underset{\mathbf{x}}{\text{argmin}} \;\; \{g(\mathbf{x}) + \frac{L}{2}\|\mathbf{x} - (\mathbf{y} - \frac{1}{L}\nabla f(\mathbf{y}))\|^2\}$$

At the kth iteration, the key step of the algorithm for solving problem (2.8) is given by

$$\mathbf{x}_k = p_L(\mathbf{x}_{k-1}) \tag{2.9}$$

where $1/L$ plays the role of a step-size. The algorithmic steps are presented below. In what follows, we refer this general method to the iterative shrinkage-thresholding algorithm (ISTA).

Algorithm 2.1 ISTA

```
Input: $L$, the Lipschitz constant of $\nabla f$.
Step 0: Take $\mathbf{x}_0 \in \mathbb{R}^n$.
Step k: ($k \geq 1$) Compute
        $\mathbf{x}_k = p_L(\mathbf{x}_{k-1})$
```

Note that ISTA reduces to the classical gradient method when $g(\mathbf{x}) \equiv 0$. It is known that for the gradient method the sequence of function values $F(\mathbf{x}_k)$ converges to the optimal function value $F(\mathbf{x}^*)$ at a rate which is bounded from above by $O(1/k)$ – a "sublinear" rate of convergence. It can be shown that ISTA shares the same rate of convergence as stated in the following theorem.

Theorem 1 (in [8])**.** *Let $\{\mathbf{x}_k\}$ be the sequence generated by (2.9). Then for any $k \geq 1$*

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{L\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{2k}$$

where $\mathbf{x}^*$ is the minimizer of $F(\mathbf{x})$.

From the theorem it follows that the number of iterations of ISTA required to obtain an ε-optimal solution, that is, an $\mathbf{x}_k$ such that $F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \varepsilon$, is at most $\lceil L\|\mathbf{x}_0 - \mathbf{x}^*\|^2/2\varepsilon \rceil$.

ISTA and the ℓ_1-ℓ_2 Optimization Problem in (2.1)

It is not hard to observe that the ℓ_1-ℓ_2 regularization problem (2.1) is a special instance of the general problem (2.8) when we set $f(\mathbf{s}) = \|\mathbf{\Theta}\mathbf{s} - \mathbf{y}\|^2$ and $g(\mathbf{s}) = \lambda\|\mathbf{s}\|_1$. The proximal-point (P-P) function in the case of an ℓ_1-ℓ_2 problem is given by

$$Q_L(\mathbf{s}, \mathbf{s}_{k-1}) = \lambda\|\mathbf{s}\|_1 + \frac{L}{2}\left\|\mathbf{s} - \Big(\mathbf{s}_{k-1} - \frac{1}{L}\nabla f(\mathbf{s}_{k-1})\Big)\right\|^2 + \text{const} \tag{2.10}$$

where L is the smallest Lipschitz constant of ∇f, i.e., $L = 2\lambda_{\max}(\mathbf{\Theta}\mathbf{\Theta}^T)$. The kth iteration of ISTA finds the next iterate $\mathbf{s}_k$ by minimizing $Q_L(\mathbf{s}, \mathbf{s}_{k-1})$, i.e.,

$$\mathbf{s}_k = p_L(\mathbf{s}_{k-1}) = \underset{\mathbf{s}}{\operatorname{argmin}}\, Q_L(\mathbf{s}, \mathbf{s}_{k-1})$$

Because of the introduction of ℓ_1 term, both terms in $Q_L(\mathbf{s}, \mathbf{s}_{k-1})$ are coordinate-separable. It can be readily verified that the minimizer of $Q_L(\mathbf{s}, \mathbf{s}_{k-1})$ can be calculated by a simple soft shrinkage with a constant threshold λ/L as

$$\mathbf{s}_k = \mathcal{T}_{\lambda/L}\Big(\mathbf{s}_{k-1} - \frac{1}{L}\nabla f(\mathbf{s}_{k-1})\Big)$$

where operator $\mathcal{T}$ applies to a vector pointwisely with $\mathcal{T}_a(\cdot) = \text{sign}(\cdot)\max\{|\cdot| - a, 0\}$ [8]. Once iterate $\mathbf{s}_k$ is obtained, it is used to obtain the next iterate by shrinkage. The iteration continues until certain stopping criterion is met.

2.4 Fast Iterative Shrinkage-Thresholding Algorithm

Evidently, the complexity of ISTA is quite low. However, the algorithm only provides a slow convergence rate of $O(1/k)$. A new proximal-point-function based algorithm, known as the fast iterative shrinkage-thresholding algorithm (FISTA), is proposed in [8, 9]. It is shown that FISTA provides a much improved convergence rate of $O(1/k^2)$ whereas the complexity of each iteration is practically the same as that of ISTA. The steps in the kth iteration of FISTA are outlined in Algorithm 2.2.

Algorithm 2.2 FISTA (in [8])

```
Input: $L$, the Lipschitz constant of $\nabla f$.
Step 0: Take $\mathbf{y}_1 = \mathbf{x}_0 \in \mathbb{R}^n$ and $t_1 = 1$.
Step $k$: $(k \geq 1)$ Compute
    $$\begin{aligned} \mathbf{x}_k &= p_L(\mathbf{y}_k) \\ t_{k+1} &= \frac{1+\sqrt{1+4t_k^2}}{2} \\ \mathbf{y}_{k+1} &= \mathbf{x}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}_k - \mathbf{x}_{k-1}) \end{aligned}$$
```

We see that the FISTA is built on ISTA with an extra step in each iteration that, with the help of a sequence of scaling factors t_k, creates an auxiliary iterate $\mathbf{y}_{k+1}$ by moving the current iterate $\mathbf{x}_k$ along the direction of $\mathbf{x}_k - \mathbf{x}_{k-1}$ so as to improve the subsequent iterate $\mathbf{x}_{k+1}$. In each round of iteration, the main computational effort in both ISTA and FISTA remains the same while the requested additional computation to obtain t_{k+1} and $\mathbf{y}_{k+1}$ is quite light. A much improved convergence rate of $O(1/k^2)$ for FISTA is established in the following theorem.

Theorem 2 (in [8])**.** *Let $\{\mathbf{x}_k\}$, $\{\mathbf{y}_k\}$ be generated by FISTA. Then for any $k \geq 1$*

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{2L\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{(k+1)^2}$$

where $\mathbf{x}^*$ is the minimizer of $F(\mathbf{x})$.

In other words, the number of iterations required by FISTA to obtain an ε-optimal solution, that is, an $\mathbf{x}_k$ such that $F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \varepsilon$, is at most $\lceil \sqrt{2L\|\mathbf{x}_0 - \mathbf{x}^*\|^2/\varepsilon} - 1\rceil$. Clearly, this is a much improved result over ISTA.

Furthermore, by including an additional step to FISTA, the algorithm is enhanced to possess desirable monotone convergence [9]. The modified algorithm is known as the monotone FISTA or MFISTA, which is presented in Algorithm 2.3. It turns out that MFISTA possesses the same convergence rate $O(1/k^2)$ as that for FISTA, see [9] for a detailed proof.

Algorithm 2.3 MFISTA (in [9])

```
**Input:** $L$, the Lipschitz constant of $\nabla f$.
**Step 0:** Take $\mathbf{y}_1 = \mathbf{x}_0 \in \mathbb{R}^n$ and $t_1 = 1$.
**Step $k$:** ($k \geq 1$) Compute
    $$\begin{aligned} \mathbf{z}_k &= p_L(\mathbf{y}_k) \\ t_{k+1} &= \frac{1+\sqrt{1+4t_k^2}}{2} \\ \mathbf{x}_k &= \text{argmin}\,\{F(\mathbf{x}) : \mathbf{x} = \mathbf{z}_k, \mathbf{x}_{k-1}\} \\ \mathbf{y}_{k+1} &= \mathbf{x}_k + \left(\frac{t_k}{t_{k+1}}\right)(\mathbf{z}_k - \mathbf{x}_k) + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}_k - \mathbf{x}_{k-1}) \end{aligned}$$
```

FISTA and the ℓ_1-ℓ_2 Optimization Problem in (2.1)

By setting $f(\mathbf{s}) = \|\mathbf{\Theta s} - \mathbf{y}\|^2$ and $g(\mathbf{s}) = \lambda\|\mathbf{s}\|_1$ in the general problem (2.8), FISTA applies to the ℓ_1-ℓ_2 problem in (2.1). By Algorithm 2.2, the steps in the kth iteration of FISTA as applied to the ℓ_1-ℓ_2 problem (2.1) are outlined as follows.

1. Perform shrinkage $\mathbf{s}_k = \mathcal{T}_{\lambda/L}\Big(\mathbf{b}_k - \frac{1}{L}\nabla f(\mathbf{b}_k)\Big)$;
2. Compute $t_{k+1} = \dfrac{1+\sqrt{1+4t_k^2}}{2}$;
3. Update $\mathbf{b}_{k+1} = \mathbf{s}_k + \left(\dfrac{t_k - 1}{t_{k+1}}\right)(\mathbf{s}_k - \mathbf{s}_{k-1})$.

The program starts with initial $\mathbf{b}_1 = \mathbf{s}_0$ and $t_1 = 1$ and terminates when the iteration number is greater than a prescribed integer or the ℓ_2 distance between the two most current iterates is less than a convergence tolerance.

2.5 Linearized Bregman Algorithm

As introduced in Sec. 2.2, a central problem in compressive sensing [21, 22, 47] is the recovery of a sparse signal from a relatively small number of linear measurements. A successful approach in the current CS theory deals with this signal reconstruction problem by means of nonsmooth convex programming (NCP). A representative formulation in the NCP setting examines the equality constrained problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad J(\mathbf{x}) \tag{2.11a}$$

$$\text{subject to:} \quad \mathbf{Ax} = \mathbf{b} \tag{2.11b}$$

where $J(\mathbf{x})$ is a continuous but non-differentiable objective function.

Concerning the computational aspects of the problem, a rich variety of algorithms is now available. In particular, when $J(\mathbf{x}) = \|\mathbf{x}\|_1$, (2.11) can be solved by linear programming (LP) for real-valued data or by second-order cone programming (SOCP) for complex-valued data [4]. Reliable LP and SOCP solvers are available, but they are not tailored for CS problems involving large-scale data such as digital images.

Another representative NCP formulation (2.11) with $J(\mathbf{x}) = \|\mathbf{x}\|_1$ is associated with the unconstrained ℓ_1-ℓ_2 problem (see Sec. 2.1)

$$\underset{\mathbf{x}}{\text{minimize}} \;\; \lambda\|\mathbf{x}\|_1 + \|\mathbf{Ax} - \mathbf{b}\|^2 \tag{2.12}$$

where $\|\cdot\|$ denotes the ℓ_2 norm and parameter λ regularizes signal sparsity while taking signal fidelity into account. Gradient-based algorithms that are especially suited for large-scale CS problems have been developed [124]. Of particular interest are those based on proximal-point functions in conjunction with iterative shrinkage techniques. These include the fast iterative shrinkage-thresholding algorithm (FISTA) and monotone FISTA (MFISTA) [8], which have been discussed in Sec. 2.4. A problem with these algorithms is that, for a solution of (2.12) to be a good approximate solution of (2.11), parameter λ in (2.12) must be sufficiently small that inevitably slows down the FISTA as a large number of iterations are required for the algorithm to converge.

In [19, 119, 120], solution methods for problem (2.11) based on the concept of Bregman distance [12] are proposed. These methods are known as linearized Bregman (LB) algorithms that are suited for large-scale problems and shown to be able to identify global minimizer of (2.11) efficiently. In addition, the LB algorithm is shown to be equivalent to a gradient descent algorithm applied to a dual formulation [119].

Bregman iteration uses Bregman Distance for finding extrema of convex functionals [12] in functional analysis, and was first applied in image processing in [89]. It has also been applied to solve the basis pursuit problem in [19, 90, 120] for compressive sensing and sparse denoising, and medical imaging problems in [34]. It is also established in [119, 120] that the original Bregman method is equivalent to the augmented Lagrangian method (the method of multipliers) [66, 96]. The Bregman distance [12] with respect to a convex function $J(\cdot)$ between points $\mathbf{u}$ and $\mathbf{v}$ is defined as

$$D_J^{\mathbf{p}}(\mathbf{u},\mathbf{v}) = J(\mathbf{u}) - J(\mathbf{v}) - \langle \mathbf{p}, \mathbf{u}-\mathbf{v}\rangle \tag{2.13}$$

where $\mathbf{p} \in \partial J(\mathbf{v})$, the subdifferential [4] of J at $\mathbf{v}$. Apparently, this is not a distance in the usual sense because it is not in general symmetric. On the other hand, it does measure the closeness between $\mathbf{u}$ and $\mathbf{v}$ in the sense that $D_J^{\mathbf{p}}(\mathbf{u},\mathbf{v}) \geq 0$, and $D_J^{\mathbf{p}}(\mathbf{u},\mathbf{v}) \geq D_J^{\mathbf{p}}(\mathbf{w},\mathbf{v})$ for $\mathbf{w}$ on the line segment between $\mathbf{u}$ and $\mathbf{v}$ [60].

The linearized Bregman (LB) method proposed in [120] is a variant of the original Bregman method introduced in [89, 120]. Its convergence and optimality properties are investigated in [15] and [19]. An LB algorithm for problem (2.11) as presented in [120] is sketched below as Algorithm 2.4, where we have adopted the notation of [69] for presentation consistency.

Algorithm 2.4 LB ([120])

```
Input: $\mathbf{x}^0 = \mathbf{p}^0 = \mathbf{0}$, $\mu > 0$ and $\tau > 0$.
for $k = 0, 1, ...K$ do
    $\mathbf{x}^{k+1} = \text{argmin}_{\mathbf{x}}\{D_J^{\mathbf{p}^k}(\mathbf{x},\mathbf{x}^k) + \tau\langle \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b}), \mathbf{x}\rangle + \frac{1}{2\mu}\|\mathbf{x}-\mathbf{x}^k\|^2\}$;
    $\mathbf{p}^{k+1} = \mathbf{p}^k - \tau\mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b}) - \frac{1}{\mu}(\mathbf{x}^{k+1}-\mathbf{x}^k)$;
end for
```

Several important results of the LB method are summarized below as Propositions 1 and 2.

Proposition 1 (in [19])**.** *Suppose $J(\cdot)$ is convex and continuously differentiable, and its gradient satisfies*

$$\|\nabla J(\mathbf{u}) - \nabla J(\mathbf{v})\|^2 \leq \beta\langle \nabla J(\mathbf{u}) - \nabla J(\mathbf{v}), \mathbf{u}-\mathbf{v}\rangle \tag{2.14}$$

for $\forall \mathbf{u},\mathbf{v} \in \mathbb{R}^N$. Then the sequence $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ generated by Algorithm 2.4 with $0 < \tau <$

$\frac{2}{\mu\|\mathbf{A}\mathbf{A}^T\|}$ *converges. The limit of* $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ *is the unique solution of*

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & J(\mathbf{x}) + \frac{1}{2\mu}\|\mathbf{x}\|^2 && \text{(2.15a)} \\ \text{subject to :} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} && \text{(2.15b)} \end{aligned}$$

Note that if μ is sufficiently large, problem (2.15) is basically equivalent to (2.11) such that Algorithm 2.4 is able to converge to the global minimizer of (2.11). However, Proposition 1 is not applicable when $J(\cdot) = \|\cdot\|_1$ because the ℓ_1-norm is not differentiable. For the ℓ_1-norm case, we have the following proposition.

Proposition 2 (in [15])**.** *Let* $J(\cdot) = \|\cdot\|_1$. *Then the sequence* $\{\mathbf{x}^k\}_{k\in\mathbb{N}}$ *generated by Algorithm 2.4 with* $0 < \tau < \frac{1}{\mu\|\mathbf{A}\mathbf{A}^T\|}$ *converges to the unique solution of problem (2.15). Let* $\mathcal{S}$ *be the set of all solutions of problem (2.11) when* $J(\mathbf{x}) = \|\mathbf{x}\|_1$ *and define* $\mathbf{x}_1$ *as the unique minimum* ℓ_2*-norm solution among all the solutions in* $\mathcal{S}$*, i.e.,* $\mathbf{x}_1 = \text{argmin}_{\mathbf{x}\in\mathcal{S}}\|\mathbf{x}\|^2$. *Denote the solution of (2.15) to be* $\mathbf{x}_\mu$. *Then* $\|\mathbf{x}_\mu\| \leq \|\mathbf{x}_1\|$ *for all* $\mu > 0$ *and* $\lim_{\mu\to\infty}\|\mathbf{x}_\mu - \mathbf{x}_1\| = 0$.

Proposition 2 is introduced and proved as the main theorem in [15]. It demonstrates that for non-differentiable function $J(\cdot) = \|\cdot\|_1$, the linearized Bregman algorithm still converges to the unique solution of problem (2.15), which is essentially the solution of (2.11) that has the minimal ℓ_2-norm among all the solutions of (2.11).

2.6 Total Variation Regularized Problems

In this section we introduce total variation (TV) regularized problems for image denoising, deblurring and compressive imaging. Investigated by Rudin, Osher and Fatemi (ROF) in [98], the total variation model is a regularization approach capable of handling edges properly and has been successful in a wide range of applications in image processing. The TV-based model is formulated, in general terms, as an unconstrained convex minimization problem of the form

$$\underset{\mathbf{U}}{\text{minimize}} \;\; \text{TV}(\mathbf{U}) + \frac{\mu}{2}\|\mathcal{A}(\mathbf{U}) - \mathbf{B}\|_{\text{F}}^2 \tag{2.16}$$

where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm, $\mathbf{B} \in R^{m\times n}$ is the observed data and $\mathbf{U} \in R^{m\times n}$ denotes the desired unknown image to be recovered. The operator $\mathcal{A}$ is a linear map and has various representations in different applications. For instance, in an image denoising setting, $\mathcal{A}$ simply corresponds to the identity operator, whereas $\mathcal{A}$ represents some blurring operator in the case of an image deblurring setting.

A great deal of research has been focused on developing efficient methods to solve (2.16). Variants of the ROF algorithm with improved performance and complexity are now available [9, 28, 31]. In particular, a main focus has been on the denoising problem, where the algorithms developed often cannot be readily extended to handle deblurring or compressive imaging problems that are more involved. On the other hand, the literature abounds on numerical methods for solving (2.16), including partial differential equation (PDE) and fixed point techniques, primal-dual Newton-based methods, primal-dual active methods, interior point algorithms and second-order cone programming, see [28–30, 33, 59, 67, 109] and the references therein.

The discretized anisotropic and isotropic TV of image $\mathbf{U}$ are defined as [9]

$$\mathrm{TV}^{(\mathrm{A})}(\mathbf{U}) = \sum_{i=1}^{m-1}\sum_{j=1}^{n} |U_{i,j} - U_{i+1,j}| + \sum_{i=1}^{m}\sum_{j=1}^{n-1} |U_{i,j} - U_{i,j+1}| \tag{2.17}$$

and

$$\begin{aligned}\mathrm{TV}^{(\mathrm{I})}(\mathbf{U}) = &\sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \sqrt{|U_{i,j} - U_{i+1,j}|^2 + |U_{i,j} - U_{i,j+1}|^2} \\ &+ \sum_{i=1}^{m-1} |U_{i,n} - U_{i+1,n}| + \sum_{j=1}^{n-1} |U_{m,j} - U_{m,j+1}|\end{aligned} \tag{2.18}$$

respectively.

Image Denoising by TV Minimization

Image denoising is probably the most successful application of TV minimization [98]. Let the image model be given by

$$\mathbf{B} = \mathbf{U}^* + \mathbf{W} \tag{2.19}$$

where $\mathbf{B}$ denotes noisy measurement of desired image $\mathbf{U}^* \in \mathbf{R}^{m\times n}$ and $\mathbf{W}$ is the noise term with independently and identically distributed (i.i.d.) Gaussian entries of zero mean and variance σ^2. The denoising of $\mathbf{B}$ is carried out by solving the convex

TV-regularized problem

$$\underset{\mathbf{U}}{\text{minimize}} \ \ \mathrm{TV}(\mathbf{U}) + \frac{\mu}{2}\|\mathbf{U} - \mathbf{B}\|_{\mathrm{F}}^2 \tag{2.20}$$

where $\mu > 0$ is a regularization parameter. Clearly, model (2.16) recovers (2.20) when $\mathcal{A}$ is taken to be the identity operator.

Compressive Imaging by TV Minimization

Compressive Sensing (CS) is now well known for more effective signal reconstruction using fewer samples, compared with the conventional Nyquist sampling. One of its significant achievements is its application in magnetic resonance imaging (MRI), due to its capability of producing high quality images with reduced imaging time. Consequently, efficient algorithms for this problem are extremely desirable.

Suppose we want to recover an MRI image $\mathbf{U} \in \mathbb{R}^{n\times n}$ based on randomized Fourier samples. If TV is used as the sparsifying transform, the optimization model can be expressed as

$$\begin{aligned} \underset{\mathbf{U}}{\text{minimize}} \quad & \mathrm{TV}(\mathbf{U}) & \text{(2.21a)} \\ \text{subject to:} \quad & \|\mathbf{R} \circ (\mathcal{F}\mathbf{U}) - \mathbf{B}\|_{\mathrm{F}}^2 < \sigma^2 & \text{(2.21b)} \end{aligned}$$

where $\mathcal{F}$ denotes the 2-D Fourier transform operator, $\mathbf{R}$ represents a random sampling matrix whose entries are either 1 or 0, $\mathbf{B}$ stores the compressive sampled measurements, and symbol $\circ$ denotes the Hadamard product or the entrywise product between two matrices. Problem (2.21) is a general formulation for sparse MRI reconstruction as presented and discussed in [34, 71, 78]. It is important to note that unlike other formulations, (2.21) deals with matrix variables that facilitates efficient analysis and fast computation as will be demonstrated later in the thesis.

Chapter 3

Methods for ℓ_p-ℓ_2 Problems

The results reported in this chapter are related to a nonconvex extension of the popular ℓ_1-ℓ_2 formulation (presented in Sec. 2.1) in the general area of sparse signal processing. The specific nonconvex problem we propose to solve is an ℓ_p-ℓ_2 problem with $0 \leq p < 1$. A fast solver for global minimization of the problem in case of an orthogonal basis is devised. Built on a recent algorithm, known as the (monotone) fast iterative shrinkage/thresholding algorithm (FISTA/MFISTA), we are able to develop algorithms for solving the ℓ_p-ℓ_2 problem where an overcomplete dictionary is adopted. The key ingredient of the algorithm is a parallel global solver that replaces the soft shrinkage within FISTA/MFISTA. Due to the nonconvex nature of the problem, we develop a power-iterative strategy for the local algorithms to reach solutions which are likely globally optimal. We also present experimental studies that evaluate the performance of the proposed techniques for signal sparse representation and compressive sensing. In the second part of the chapter, we present a practical signal denosing technique by virtue of the ℓ_p norm. A smoothed ℓ_p-ℓ_2 solver is proposed to deal with the oscillations that often occur in the ℓ_p SNR profiles when the conventional global solver is employed. The usefulness of the algorithm is demonstrated by simulations in denoising a variety of 1-D and 2-D signals.

3.1 Fast Iterative Algorithm for ℓ_p-ℓ_2 Optimization

A *nonconvex* variant of the basis pursuit (BP) problem can be formulated by replacing the ℓ_1 norm term in BP with an ℓ_p norm with $0 < p < 1$ [35, 36]. The ℓ_p norm of vector $\mathbf{s}$ is defined as $\|\mathbf{s}\|_p = (\sum_{i=1}^{N} |s_i|^p)^{1/p}$. We remark that with $p < 1$, the "ℓ_p

norm" is no longer a norm because it does not satisfy the triangle inequality condition required to be a norm, however $\|\mathbf{s}\|_p^p = \sum_{i=1}^N |s_i|^p$ satisfies the triangle inequality and remains to be a meaningful distance measure. It was demonstrated by numerical experiments [35] that fewer measurements than that of BP are required by the ℓ_p-ℓ_2 BP for exact reconstruction of a sparse signal. Naturally, an ℓ_p-ℓ_2 problem can be formulated as an unconstrained problem:

$$\text{minimize } \ F(\mathbf{s}) = \lambda||\mathbf{s}||_p^p + ||\boldsymbol{\Theta}\mathbf{s} - \mathbf{y}||^2 \tag{3.1}$$

where $\mathbf{s} \in R^N$, $\boldsymbol{\Theta} \in R^{M\times N}$ and $\mathbf{y} \in R^M$.

Our algorithm for (3.1) is built on a recent algorithm, known as the fast iterative shrinkage/thresholding algorithm (FISTA) [8], where the key soft shrinkage step is replaced by a new solver for global minimization of a 1-D nonconvex ℓ_p problem. Unlike a typical ℓ_1-ℓ_2 proximal-point (P-P) objective function [124], we associate each iteration of our algorithm to a P-P objective function given by

$$Q_p(\mathbf{s}, \mathbf{b}_k) = \lambda\|\mathbf{s}\|_p^p + \frac{L}{2}\|\mathbf{s} - \Big(\mathbf{b}_k - \frac{1}{L}\nabla f(\mathbf{b}_k)\Big)\|^2 \tag{3.2}$$

where $f(\mathbf{s}) = \|\boldsymbol{\Theta}\mathbf{s} - \mathbf{y}\|^2$. At a glance, function $Q_p(\mathbf{s}, \mathbf{b}_k)$ differs from $Q_1(\mathbf{s}, \mathbf{b}_k)$ only slightly with its ℓ_1 term replaced by an ℓ_p term. However, this change turns out to be a rather major one in several aspects. On one hand, with $p < 1$ (3.2) provides a problem setting closer to the ℓ_0-norm minimization, where the ℓ_0-norm denotes the number of nonzero elements of the vector. The ℓ_0-norm based signal recovery is attractive as it can facilitate exact recovery of sparse signal. Consequently, with the ℓ_p variation for p less than 1, improved sparse signal recovery performance is expected. And this is indeed the very reason of the studies reported in this chapter. On the other hand, with $p < 1$ the problem in (3.2) becomes nonconvex, hence conventional technique like soft shrinkage fails to work in general, and this technical difficulty motivates the development of a new solver for problem (3.2).

For notational simplicity, the problem of minimizing $Q_p(\mathbf{s}, \mathbf{b}_k)$ can be cast as

$$\text{minimize } \lambda||\mathbf{s}||_p^p + \frac{L}{2}||\mathbf{s} - \mathbf{c}||^2 \tag{3.3}$$

where $\mathbf{c} = \mathbf{b}_k - \frac{1}{L}\nabla f(\mathbf{b}_k)$. With $p < 1$, the minimization problem (3.3) is nonconvex. By taking advantage of the objective function in (3.3) being separable in coordinates of $\mathbf{s}$, below we devise a fast parallel solver [115, 116] to secure the global minimizer of (3.3), then incorporate this solver into an FISTA framework by replacing the conventional soft shrinkage operator.

3.1.1 A Parallel Global Solver for the ℓ_p-ℓ_2 Problem (3.3)

The objective function in (3.3) consists of two terms, both of which are separable. Consequently, (3.3) is reduced to a series of N 1-D problems of the form

$$\text{minimize } u(s) = \lambda|s|^p + \frac{L}{2}(s-c)^2 \tag{3.4}$$

In the following, we first present an algorithm for finding the global solution s^* of (3.4) [116].

Global Solver for the 1-D Problem (3.4)

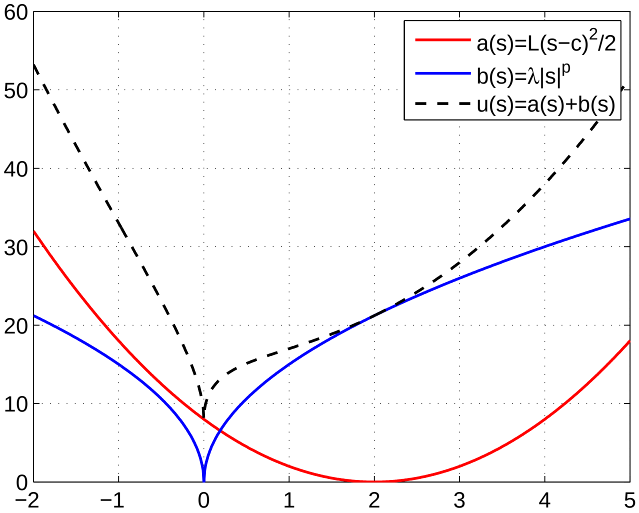


Figure 3.1: Function $u(s)$ with $c > 0$.

To begin with, we examine function $u(s)$ with respect to parameter c. If $c = 0$, it is obvious that $s^* = 0$. Next, we consider the case of $c > 0$. To illustrate the current circumstance, Fig. 3.1 plots $a(s) = \dfrac{L}{2}(s-c)^2$, $b(s) = \lambda|s|^p$ and $u(s) = a(s) + b(s)$ for some L, c, λ and p. It can be observed that when variable s is in the region $(-\infty, 0)$, functions $a(s)$ and $b(s)$ are both monotonically decreasing; in addition when

$s \in (c, +\infty)$, $a(s)$ and $b(s)$ are both monotonically increasing. Hence the global minimizer s^* lies in $[0, c]$ where the function of interest becomes

$$u(s) = \lambda s^p + \frac{L}{2}(s-c)^2 \quad \text{for } s \in [0, c] \tag{3.5}$$

As mentioned earlier, gradient information is not sufficient to identify the global minimizer. The convexity property of function $u(s)$ can be analyzed by examining the 2nd-order derivative of (3.5), i.e.,

$$u''(s) = L + \lambda p(p-1)s^{p-2} \tag{3.6}$$

By solving the equation $u''(s) = 0$, we obtain

$$s_c = \left[\frac{\lambda p(1-p)}{L}\right]^{1/(2-p)}$$

Clearly, $s_c > 0$. For $0 \leq s < s_c$, $u(s)$ is concave as $u''(s) < 0$; for $s > s_c$, $u(s)$ is convex as $u''(s) > 0$. For s in interval $[0, c]$, two cases need to be examined.

1. If $s_c \geq c$, $u(s)$ is concave in $[0, c]$. As a result, s^* must be either 0 or c. Namely, $s^* = \text{argmin } \{u(s) : s = 0, c\}$. This case is illustrated in Fig. 3.2. From $s_c \geq c$, it can be derived that $Lc^{2-p} \leq \lambda p(1-p) \leq \lambda/4$, which further gives $\frac{L}{2}c^2 < 4Lc^2 \leq \lambda c^p$, i.e., $u(0) < u(c)$. Therefore, $s^* = 0$ for $s_c \geq c$.
2. If $s_c < c$, as illustrated in Fig. 3.3, $u(s)$ is concave in $[0, s_c]$ and convex in $[s_c, c]$. More specifically, if $u'(s_c) \geq 0$, $u(s)$ is monotonically non-decreasing in the interval $[0, c]$, thus $s^* = 0$. On the other hand if $u'(s_c) < 0$, we argue that s^* must be either at the boundary point 0, or at the point s_t that minimizes convex function $u(s)$ in $[s_c, c]$. Point s_t can be identified by several rounds of iterations based on bisection search method. Hence the global minimizer is obtained as $s^* = \text{argmin } \{u(s) : s = 0, s_t\}$.

Based on this, a global solver of (3.4) for $c \geq 0$ can readily be generated. If we denote the solution of this solver by $s^* = \texttt{gsol}(c, L, \lambda, p)$, then it is evident that the global solution of (3.4) for $c < 0$ can be obtained as $s^* = -\texttt{gsol}(-c, L, \lambda, p)$.

In spite of its plain structure, a drawback of this solution method is its low efficiency, especially for large-scale problems, as one needs to solve N 1-D problems in

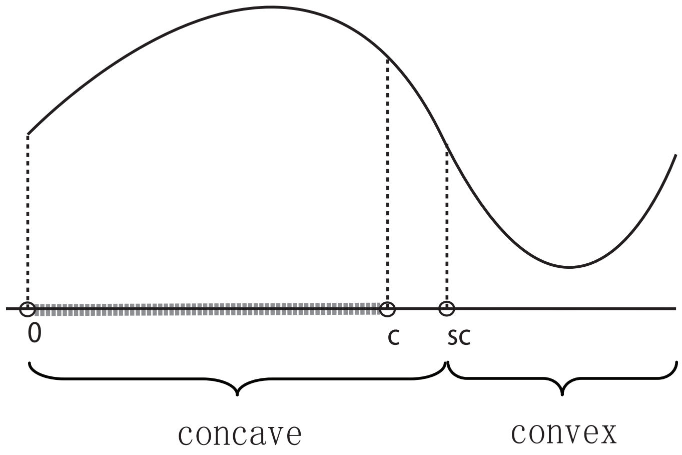


Figure 3.2: Function $u(s)$ when $s_c \geq c$ (with $c > 0$).

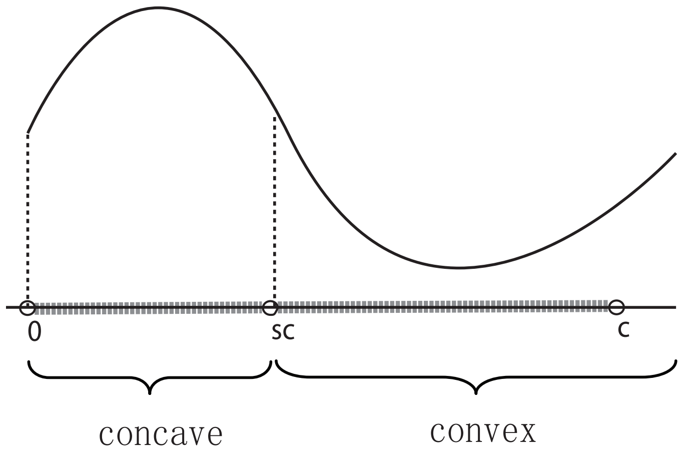


Figure 3.3: Function $u(s)$ when $s_c < c$ (with $c > 0$).

order to minimize (3.3). Below we describe an improved algorithm which employs a parallel processing technique to accelerate the global solver.

Fast Implementation of the Global Solver for Problem (3.3)

The following notations are adopted. We denote by $\mathbf{a} \circ \mathbf{b}$ the component-wise product of vectors $\mathbf{a}$ and $\mathbf{b}$, by $\mathbf{a}.^p$ the vector whose ith component is $|a_i|^p$, and by $\mathbf{1}$ and $\mathbf{0}$ the all-one and all-zero vectors, respectively. Let Λ be a length-K subset of $\{1, 2, ..., N\}$, $\mathbf{c}$ be a vector of length N and $\mathbf{b}$ be a vector of length K. We use $\mathbf{c}(\Lambda)$ to denote a vector of length K that retains those components of $\mathbf{c}$ whose indices are in Λ; $\mathbf{c}(\Lambda) = \mathbf{b}$ to denote a vector of length N obtained by updating the components of $\mathbf{c}$, whose indices are in Λ, with those of $\mathbf{b}$; $\mathbf{d} = \mathbf{c}(\Lambda)$ denotes a vector of length K that retains those components of $\mathbf{c}$ whose indices are in Λ. We use $[\mathbf{a} > \mathbf{b}]$ ($[\mathbf{a} < \mathbf{b}]$) to denote a vector whose ith component is 1 if $a_i > b_i$ ($a_i < b_i$) and 0 otherwise; and $[\mathbf{a} \geq \mathbf{b}]$ ($[\mathbf{a} \leq \mathbf{b}]$) is similarly defined.

A step-by-step description of the algorithm through a parallel implementation is described in Table 3.1 as Algorithm 3.1 where it is quite obvious that the data are processed in a vector-wise rather than component-wise manner. The parallel processing of data is made possible by taking the advantage of the separable structure of the objective function in (3.3) and playing a technical trick about the signs of $\mathbf{c}$ as illustrated for the scalar case earlier. In particular, the signs of $\mathbf{c}$ are recorded into a vector $\boldsymbol{\theta}$, which is used to accommodate the negative-sign cases simply by a component-wise product at the last step.

Most of the steps in Algorithm 3.1 can be parallel implemented in MATLAB based on vector operations so that "for loops" are avoided as much as possible. We remark that the proposed ℓ_p-ℓ_2 solver is highly parallel with exception only in Step 3.4 where a total of $|\Omega|$ calls for bisection search are made. Since $|\Omega|$ is typically much smaller than N, overall the complexity of the proposed solver is considerably reduced compared with that required by applying 1-D solver `gsol` N times.

3.1.2 Performance of the Parallel ℓ_p-ℓ_2 Solver for Denoising a 1-D Signal with Orthogonal Basis

In this section, problem (3.1) is investigated with an orthogonal $\boldsymbol{\Theta}$, i.e., $\boldsymbol{\Theta}\boldsymbol{\Theta}^T = \boldsymbol{\Theta}^T\boldsymbol{\Theta} = \mathbf{I}$. Then with

$$\|\boldsymbol{\Theta}\mathbf{s} - \mathbf{y}\|^2 = \|\boldsymbol{\Theta}(\mathbf{s} - \boldsymbol{\Theta}^T\mathbf{y})\|^2 = \|\mathbf{s} - \boldsymbol{\Theta}^T\mathbf{y}\|^2$$

we write the objective function in (3.1) as

$$F(\mathbf{s}) = \lambda\|\mathbf{s}\|_p^p + \|\mathbf{s} - \mathbf{c}\|^2. \tag{3.7}$$

where $\mathbf{c} = \boldsymbol{\Theta}^T\mathbf{y}$. Evidently in case of an orthogonal basis, global solution of (3.1) can be easily identified by minimizing (3.7) using Algorithm 3.1. We carry out an experiment for a 1-D signal denoising with orthogonal basis in the following.

A test signal of length $N = 256$ known as "`HeaviSine`" [50] (with maximum amplitude normalized to 1) was corrupted with additive white Gaussian noise $\mathbf{w}$ with zero

Algorithm 3.1	
Input Data	$\mathbf{c}$, L, λ and p.
Output Data	$\mathbf{s}^* = \text{argmin}\left\{\lambda\lVert\mathbf{s}\rVert_p^p + \frac{L}{2}\lVert\mathbf{s}-\mathbf{c}\rVert^2\right\}$.
Step 1	Set $\boldsymbol{\theta} = \text{sign}(\mathbf{c})$ and $\mathbf{c} = \boldsymbol{\theta}\circ\mathbf{c}$.
Step 2	If $p = 0$, compute $\boldsymbol{\vartheta} = \left[\frac{L}{2}\mathbf{c}.^2 > (\lambda\cdot\mathbf{1})\right]$, set $\mathbf{s}^* = \mathbf{c}\circ\boldsymbol{\vartheta}$ and do Step 4; otherwise do Step 3.
Step 3	1. Compute $s_c = [\lambda p(1-p)/L]^{1/(2-p)}$, set $\boldsymbol{\vartheta} = [(s_c\circ\mathbf{1}) < \mathbf{c}]$. 2. Define $\Lambda = \{i : \vartheta_i = 1\}$ and update $\mathbf{c} = \mathbf{c}(\Lambda)$. 3. Compute $\mathbf{v} = L(s_c\circ\mathbf{1} - \mathbf{c}) + \lambda p s_c^{p-1}\circ\mathbf{1}$, update $\boldsymbol{\vartheta} = [\mathbf{v} < \mathbf{0}]$. 4. Define $\Omega = \{i : \vartheta_i = 1\}$. For each $i\in\Omega$, replace ϑ_i by the minimizer of (3.4) over $[s_c, c_i]$, which can be computed through bisection search. 5. Set $\tilde{\mathbf{c}} = \mathbf{c}(\Omega)$ and $\tilde{\boldsymbol{\vartheta}} = \boldsymbol{\vartheta}(\Omega)$. Compute $\boldsymbol{\beta} = \left[\frac{L}{2}\tilde{\mathbf{c}}.^2 > \frac{L}{2}(\tilde{\boldsymbol{\vartheta}} - \tilde{\mathbf{c}}).^2 + \lambda\tilde{\boldsymbol{\vartheta}}.^p\right]$ and set $\boldsymbol{\vartheta}(\Omega) = \boldsymbol{\beta}\circ\tilde{\boldsymbol{\vartheta}}$. 6. Set $\mathbf{s}^* = \boldsymbol{\vartheta}$.
Step 4	Compute $\mathbf{s}^* = \boldsymbol{\theta}\circ\mathbf{s}^*$.

Table 3.1: A Fast ℓ_p-ℓ_2 Global Solver for Problem (3.3)

mean and standard deviation $\sigma = 0.08$. The signal-to-noise ratio (SNR) of the noisy signal was found to be 19.71dB. Matrix $\boldsymbol{\Psi}$ represents an orthogonal 8-level Daubechies wavelet D8 basis. With p fixed as one of the six values $\{1, 0.8, 0.6, 0.4, 0.2, 0\}$, the parallel ℓ_p-ℓ_2 global solver in Algorithm 3.1 was applied to obtain global solutions to problem (3.1) (where $\boldsymbol{\Theta} = \boldsymbol{\Psi}$) with uniformly placed λ from 0 to 0.1. The SNR obtained versus λ for each p are depicted as six curves in Fig. 3.4. It is observed that in most cases, using $p < 1$ offers improved SNR relative to that obtained with $p = 1$ (BPDN). The best performance was achieved with $p = 0.4$ at $\lambda = 0.1$ offering an SNR of 25.78dB. Fig. 3.5 illustrates the clean "`HeaviSine`" signal, the noise-corrupted signal, the denoised signal obtained by BPDN at $\lambda = 0.1$, and the ℓ_p denoised signal with $p = 0.4$ at $\lambda = 0.1$.

Although in general using a smaller p in the algorithm produces improved recovery, the denoising performance cannot be guaranteed. This is illustrated in Fig. 3.4 by the SNR profiles associated with $p = 0.2$ and $p = 0$. We shall deal with this technical

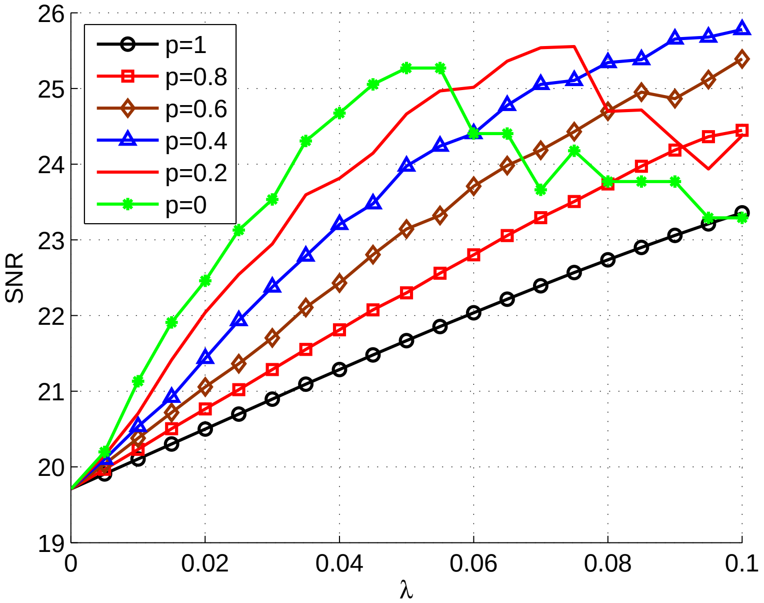


Figure 3.4: SNR produced by denoising signal "`HeaviSine`" by the parallel global ℓ_p-ℓ_2 solver with orthogonal $\mathbf{\Psi}$

difficulty in Sec. 3.2 by developing a smoothed ℓ_p-ℓ_2 solver.

3.1.3 Fast Iterative Algorithms ℓ_p-FISTA and ℓ_p-MFISTA

In previous sections, we have shown that global minimizer of $Q_p(\mathbf{s}, \mathbf{b}_k)$ can be efficiently identified using Algorithm 3.1. By replacing the conventional soft shrinkage operator with the parallel ℓ_p-ℓ_2 solver in the framework of FISTA/MFISTA, two algorithms for a (local) solution of problem (3.1) can be constructed. One of the algorithms is called the ℓ_p-FISTA, and the other, which is enhanced with the monotone convergence property, is called the ℓ_p-MFISTA. These algorithms are outlined in Tables 3.2 and 3.3 as Algorithms 3.2 and 3.3, respectively.

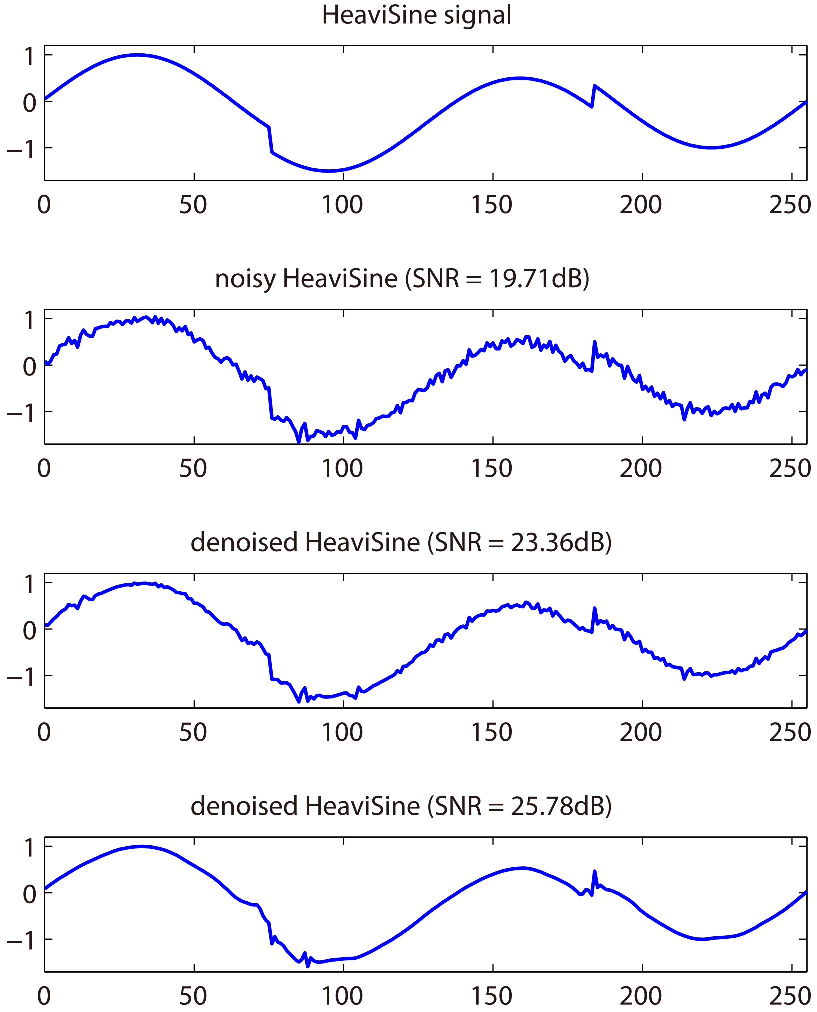


Figure 3.5: From top to bottom: original `HeaviSine`; its noisy version; denoised signal with $p = 1$ and $\lambda = 0.1$; and denoised signal with $p = 0.4$ and $\lambda = 0.1$.

A Signal Sparse Representation Example

To evaluate the fast iterative iterative algorithms for ℓ_p-ℓ_2 optimization, in this experiment we seek sparse representation for a `bumps` signal [50] $\mathbf{x}$ of length $N = 256$ as shown in Fig. 3.6. The sparse representation dictionary adopted here is a combination of three orthonormal bases $\mathbf{\Psi} = [\mathbf{\Psi}_1 \ \mathbf{\Psi}_2 \ \mathbf{\Psi}_3] \in R^{N\times 3N}$ where $\mathbf{\Psi}_1$ is the Dirac basis, $\mathbf{\Psi}_2$ is the DCT basis and $\mathbf{\Psi}_3$ is the wavelet basis generated by orthogonal Daubechies wavelet D8. Our objective is to find a representation vector $\mathbf{s} \in R^{3N\times 1}$ for signal $\mathbf{x}$ such that $\mathbf{x} \approx \mathbf{\Psi s}$ with $\mathbf{s}$ as sparse as possible. The problem can be cast as the ℓ_p-ℓ_2 problem

$$\text{minimize} \ \ \lambda||\mathbf{s}||_p^p + ||\mathbf{\Psi s} - \mathbf{x}||^2 \tag{3.8}$$

Algorithm 3.2	
Input Data	λ, p, $\mathbf{\Theta}$ and $\mathbf{y}$.
Output Data	Local solution of problem (3.1).
Step 1	Take $L = 2\lambda_{\max}(\mathbf{\Theta}\mathbf{\Theta}^T)$ as the Lipschitz constant of ∇f. Set initial iterate $\mathbf{s}_0$ and the number of iterations K_m. Set $\mathbf{b}_1 = \mathbf{s}_0$, $k = 1$ and $t_1 = 1$.
Step 2	Compute the global minimizer $\mathbf{s}_k$ of $Q_p(\mathbf{s}, \mathbf{b}_k)$ using the parallel ℓ_p-ℓ_2 solver. Then update $\begin{aligned} t_{k+1} &= (1 + \sqrt{1 + 4t_k^2})/2, \\ \mathbf{b}_{k+1} &= \mathbf{s}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{s}_k - \mathbf{s}_{k-1}) \end{aligned}$
Step 3	If $k = K_m$, stop and output $\mathbf{s}_k$ as the solution; otherwise set $k = k + 1$ and repeat from Step 2.

Table 3.2: The ℓ_p-FISTA

Algorithm 3.3	
Input Data	λ, p, $\mathbf{\Theta}$ and $\mathbf{y}$.
Output Data	Local solution of problem (3.1).
Step 1	Take $L = 2\lambda_{\max}(\mathbf{\Theta}\mathbf{\Theta}^T)$ as the Lipschitz constant of ∇f. Set initial iterate $\mathbf{s}_0$ and the number of iterations K_m. Set $\mathbf{b}_1 = \mathbf{s}_0$, $k = 1$ and $t_1 = 1$.
Step 2	Compute the global minimizer $\mathbf{z}_k$ of $Q_p(\mathbf{z}, \mathbf{b}_k)$ using the parallel ℓ_p-ℓ_2 solver. Then update $\begin{aligned} t_{k+1} &= (1 + \sqrt{1 + 4t_k^2})/2, \\ \mathbf{s}_k &= \text{argmin}\ \{F(\mathbf{s}) :\ \mathbf{s} = \mathbf{z}_k, \mathbf{s}_{k-1}\}, \\ \mathbf{b}_{k+1} &= \mathbf{s}_k + \frac{t_k}{t_{k+1}}(\mathbf{z}_k - \mathbf{s}_k) + \frac{t_k - 1}{t_{k+1}}(\mathbf{s}_k - \mathbf{s}_{k-1}). \end{aligned}$
Step 3	If $k = K_m$, stop and output $\mathbf{s}_k$ as the solution; otherwise set $k = k + 1$ and repeat from Step 2.

Table 3.3: The ℓ_p-MFISTA

Obviously, the problem is equivalent to (3.1) up to notational changes with $\mathbf{y} = \mathbf{x}$ and $\mathbf{\Theta} = \mathbf{\Psi}$. To this end we apply the ℓ_p-MFISTA to solve problem (3.1) with $p = 1, 0.95, 0.9, 0.85, 0.8$ and 0.75, respectively. For each ℓ_p-ℓ_2 problem with a particular p, the experiment was carried out by the steps outlined below.

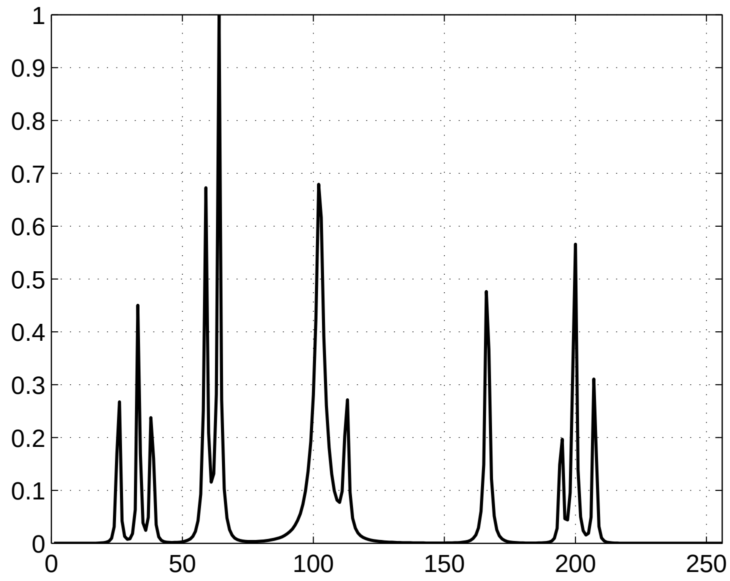


Figure 3.6: `Bumps` signal of length $N = 256$.

Step 1

Set $\mathbf{s}_0 = \mathbf{0}$ and $i = 1$. Generate a vector $\boldsymbol{\lambda} = [\lambda_1\ \lambda_2\ \cdots\ \lambda_T]$ with $\lambda_1 > \lambda_2 > \cdots > \lambda_T$. The number of iterations of the modified MFISTA was set to be $K = 200$.

Step 2

Apply the ℓ_p-MFISTA to solve problem (3.8) with initial point $\mathbf{s}_{i-1}$ and parameter $\lambda = \lambda_i$ to obtain the solution $\hat{\mathbf{s}}$. Set $\mathbf{s}_i = \hat{\mathbf{s}}$.

Step 3

Compute relative equation error

$$R_i = \frac{\|\boldsymbol{\Psi}\mathbf{s}_i - \mathbf{x}\|}{\|\mathbf{x}\|}.$$

Compute the percentage of zeros in $\mathbf{s}_i$ and denote it by Z_i (a component of $\mathbf{s}_i$ was regarded as zero if its absolute value falls below 1e-5).

Step 4

If $i = T$, stop; otherwise set $i = i + 1$ and repeat from Step 2. ■

It should be stressed that the parameter vector $\boldsymbol{\lambda}$ consists of decreasing components $\lambda_1 > \lambda_2 > \cdots > \lambda_T$. Take $p = 1$ for instance, the λ_i's were set to be an arithmetic progression from $\lambda_1 = 5\mathrm{e} - 2$ to $\lambda_T = 5\mathrm{e} - 3$ with a common difference of 5e-3. The components of $\boldsymbol{\lambda}$ were tuned for each individual value of p so that the same level of relative equation error is attained, i.e., for each individual p, we seek its solution with relative equation error on the magnitude between 1e-2 and 1e-1. As a result, for a given value of power p, two sequences $\mathbf{R} = [R_1\ R_2\ \cdots\ R_T]$ and $\mathbf{Z} = [Z_1\ Z_2\ \cdots\ Z_T]$ were produced.

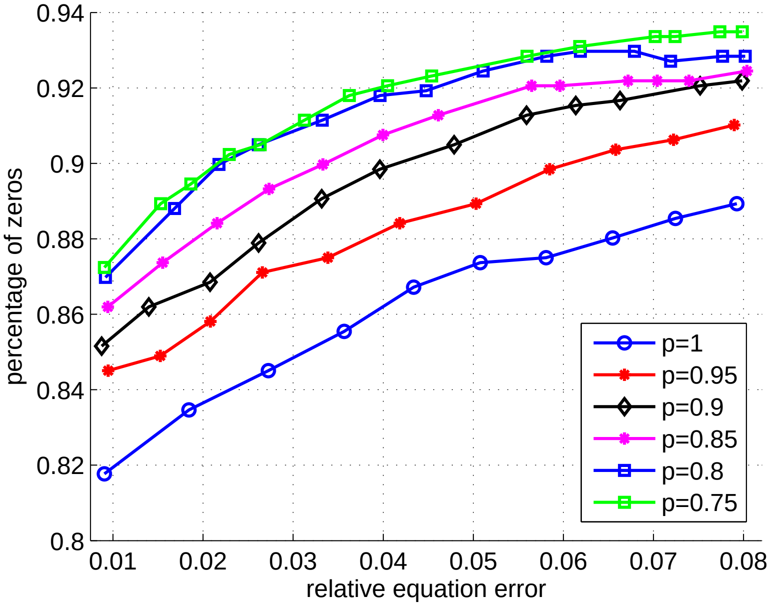


Figure 3.7: Comparison of ℓ_p-ℓ_2 sparse representation of `bumps` signal for $p = 1, 0.95, 0.9, 0.85, 0.8, 0.75$ in terms of relative equation error and signal sparsity in the dictionary domain.

The quality of a sparse representation may be evaluated by two criteria as

1. How sparse the coefficient vector $\mathbf{s}$ is in the dictionary domain;

2. How well the reconstruction $\mathbf{\Psi s}$ resembles $\mathbf{x}$.

In the experiment, sparsity was measured by computing the percentage of zeros in $\mathbf{s}$ (as seen in vector $\mathbf{Z}$), and the signal reconstruction precision was measured by the relative equation error $\|\mathbf{\Psi s} - \mathbf{x}\|/\|\mathbf{x}\|$ (as seen in vector $\mathbf{R}$). Since the value of regularization parameter λ controls the tradeoff between sparsity and equation error of the solution, a curve generated with the sparsity as its x-coordinates and the reconstruction precision as its y-coordinates provides a performance profile of the solution that shows how the sparsity/equation error evolves as parameter λ varies.

A total of six such curves for $p = 1, 0.95, 0.9, 0.85, 0.8$ and 0.75 for signal `bumps` are depicted in Fig. 3.7. From Fig. 3.7, several observations can be made:

1. For a fixed relative equation error, the sparsity improves as a smaller power p was used, and this justifies the usefulness of the proposed ℓ_p pursuit algorithm;

2. For a fixed level of sparsity, the relative equation error appears to decrease as power p decreases, a clear indication that justifies the ℓ_p pursuit algorithm;

3. The performance improvement appears to be non-proportional with respect to the change in power p. Starting from $p = 1$ (the BP pursuit), a 0.05 decrease in p leads to a significant performance improvement. As p continues to decrease, the performance continues to gain but the incremental gain becomes gradually less significant. In this simulation, the best performance is achieved at $p = 0.75$.

For further illustration, Fig 3.8 depicts the signals obtained by solving problem (3.8) with $p = 1$ and $p = 0.75$, respectively. For a fair comparison, the values of parameter λ were chosen such that both solutions yield the same relative equation error of 0.00905. Note that these two instances correspond to the two leftmost points on the two curves in Fig. 3.7 that are associated with the above two p values. The sparsity achieved was found to be 87.24% for $p = 0.75$ versus 81.77% for $p = 1$. The improvement in sparsity with $p = 0.75$ over that of $p = 1$ is visually clear in Fig. 3.8. Note that in Fig. 3.8 the components of two sparse signals are plotted over a value range of $[-0.03, 0.03]$ for better visualization.

3.1.4 A Power-Iterative Strategy for ℓ_p-ℓ_2 Optimization Towards Global Solution

Although in each iteration the ℓ_p-FISTA or ℓ_p-MFISTA minimizes the P-P function $Q_p(\mathbf{s}, \mathbf{b}_k)$ globally, a solution of problem (3.1) obtained is not guaranteed globally optimal because (3.1) is a nonconvex problem for $0 < p < 1$. In what follows we propose a power-iterative strategy that promotes a local algorithm ℓ_p-FISTA or ℓ_p-MFISTA to converge to a solution that is likely globally optimal.

The power-iterative strategy is based on the intuitive observation that for a nonconvex problem, a gradient based algorithm is not expected to converge to a global solution unless it starts at an initial point that is sufficiently close to the global solution. Specifically, for a given power $p < 1$ and an appropriate value of λ, the

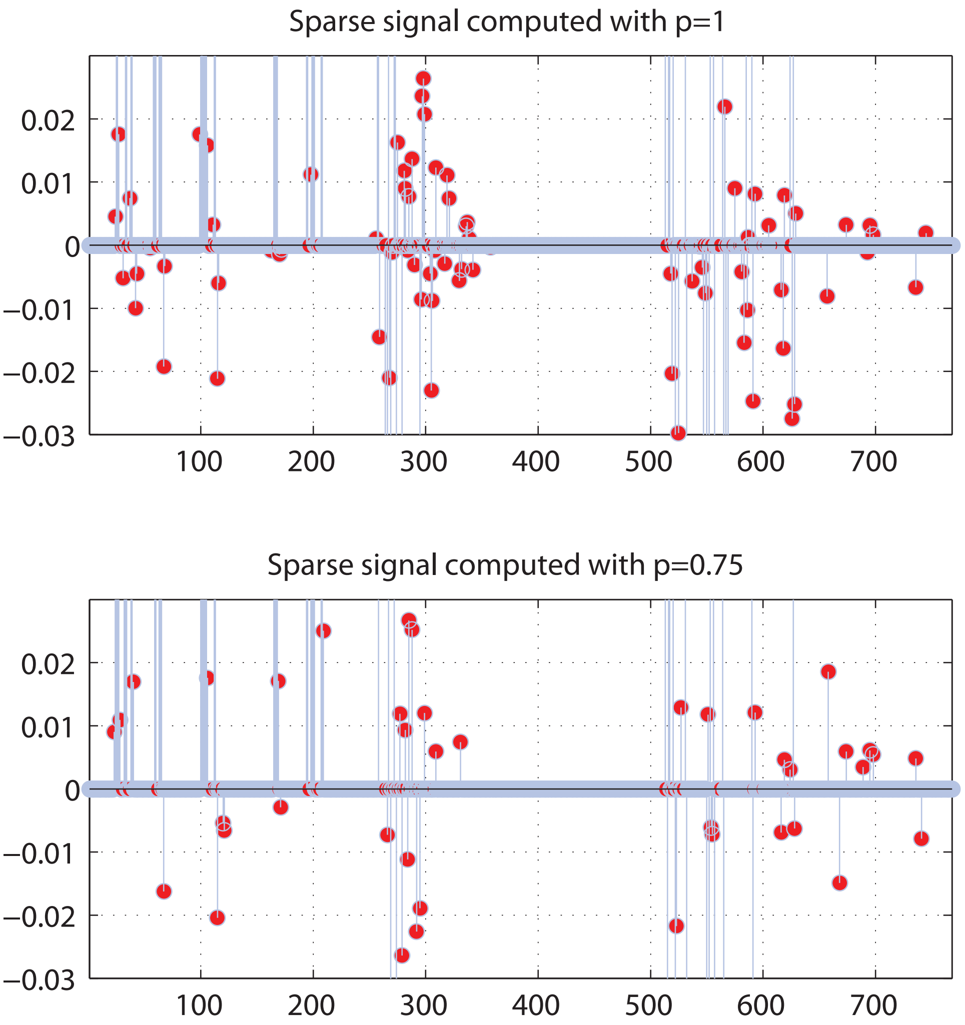


Figure 3.8: Sparse representation of the `bumps` signal based on ℓ_1 and $\ell_{0.75}$ reconstruction. Both representations yield the same relative error of 0.00905. The sparse representation computed with $p = 1$ shown in the upper graph has 81.77% zeros, while the one computed with $p = 0.75$ shown in the lower graph has 87.24% zeros.

global solutions of (3.1) associated with powers p and $p + \Delta p$ are close to each other as long as the power difference Δp is sufficiently small in magnitude. The proposed power-iterative strategy begins by solving the convex ℓ_1-ℓ_2 problem based on FISTA/MFISTA where a conventional soft-shrinkage operation is carried out in each iteration. The global solution $\mathbf{s}^{(0)}$ is then used as the initial point to start the next ℓ_p-ℓ_2 problem with a p close to but slightly less than one. This problem is solved by the ℓ_p-FISTA (or ℓ_p-MFISTA) and the solution obtained is denoted as $\mathbf{s}^{(1)}$. The iterate $\mathbf{s}^{(1)}$ is then served as an initial point for the next ℓ_p-ℓ_2 problem (3.1) with p further reduced slightly. This process continues until the target power value p_t is reached. It is worthwhile to mention an algorithm named iterative reweighted ℓ_1 (IR-L1) minimization method [84]. Empirical studies and provable results have suggested the reweighting version outperforms the standard method. On the other hand, the IRL1 essentially performs ℓ_0-regularization by jumping from ℓ_1-regularization directly to ℓ_0-regularization as opposed to the power-iterative strategy where ℓ_p regularization is implemented with a warm-start strategy and power p is reduced slightly each time. In the following, we evaluate the proposed power-iterative algorithm for signal recovery in compressive sensing.

3.1.5 Performance of the Power-Iterative Strategy on Compressive Sensing

In our simulations, each K-sparse test signal $\mathbf{s}$ was constructed by assigning K values randomly drawn from $\mathcal{N}(0, 1)$ to K randomly selected locations of a zero vector of length $N = 32$. A total of 20 values of K from 1 to 20 were used. The number of measurements was set to $M = 20$ and a measurement matrix $\mathbf{\Phi}$ of size $M \times N$ was constructed with its elements drawn from $\mathcal{N}(0, 1)$ followed by normalizing each column to unit ℓ_2 norm. Since the test signals were exactly K-sparse, we have $\mathbf{\Theta} = \mathbf{\Phi}$ in (3.1) and the power-iterative strategy in conjunction with ℓ_p-MFISTA was applied to reconstruct $\mathbf{s}$.

A sequence of power p was set from 1 to 0 with a decrement of $d = 0.1$. For each p, the ℓ_p-MFISTA was executed in a successive manner with a set of decreasing λ's such that the equality constraint was practically satisfied. A total of 50 ℓ_p-MFISTA iterations was chosen for each λ. A recovered signal $\hat{\mathbf{s}}$ was deemed perfect if the relative solution error $\|\hat{\mathbf{s}} - \mathbf{s}\|_2 / \|\mathbf{s}\|_2$ was found to be less than 1e-5. For each value of K, the number of perfect reconstructions were counted over 100 runs. Figs. 3.9 and 3.10 depict the results with $p = 1, 0.9, 0.8, 0.7, 0.4$ and 0. It is observed that

1. For a fixed sparsity K, the rate of perfect reconstruction increases and the average relative reconstruction error reduces as a smaller power p was used. This justifies the usefulness of the proposed ℓ_p pursuit algorithm;

2. The performance improvement tends to be nonlinear with respect to the change in power p, experiencing considerable improvement as p reduces from 1 to 0.9. As p decreases further, the performance continues to gain but the incremental gain becomes gradually less significant. It is also observed that the best

reconstruction performance was achieved at $p = 0$.

The reconstruction results of a K-sparse signal are illustrated in Fig. 3.11 for $K = 10$ with $p = 1$ and $p = 0$, respectively. As can be seen from the curves in Fig. 3.9, the rate of perfect reconstruction with $p = 0$ when $K = 10$ is 80%, much higher than a rate of 20% using $p = 1$. Fig. 3.11 supports this observation by a single instance, as we show that an ℓ_0-based method perfectly recovers a signal which however cannot be successfully recovered by the conventional ℓ_1-based technique.

Among other things, Figs. 3.12 and 3.13 compare the ℓ_0 (and $\ell_{0.9}$) solution obtained by the power-iterative strategy described above with an ℓ_0 (and $\ell_{0.9}$) solution obtained by ℓ_p-MFISTA with the least-squares (LS) solution or the zero vector as the initial point, showing considerable performance gain achieved by the proposed method. In particular, it can be seen that the ℓ_0 recovery performance when an LS solution is utilized as the initial point is even worse than the ℓ_1 benchmark. This suggests that choosing an adequate initial point constructed through the power iterative strategy greatly affects signal recovery performance, as different initial points lead to different local solutions. The simulations conducted so far have indicated that the proposed power-iterative method has the potential in approaching a global solution of the nonconvex problem (3.1).

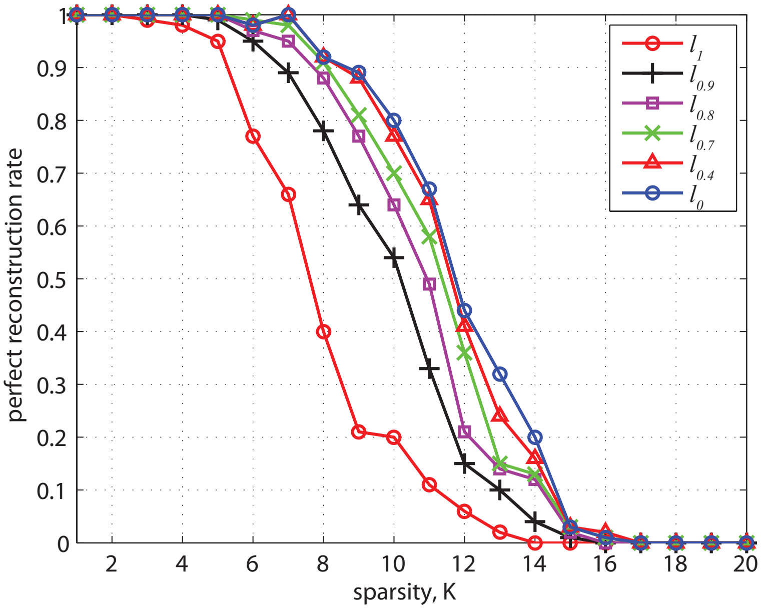


Figure 3.9: Rate of perfect reconstruction for ℓ_p-ℓ_2 problems with $p = 1, 0.9, 0.8, 0.7, 0.4$ and 0 over 100 runs for signals of length $N = 32$ and number of random measurements $M = 20$.

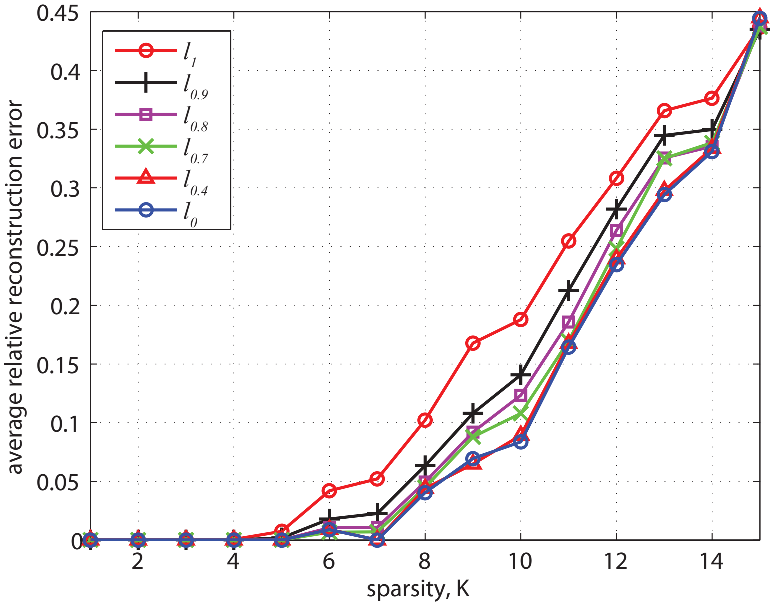


Figure 3.10: Average relative reconstruction errors for ℓ_p-ℓ_2 problems with $p = 1, 0.9, 0.8, 0.7, 0.4$ and 0 over 100 runs for signals of length $N = 32$ and number of random measurements $M = 20$.

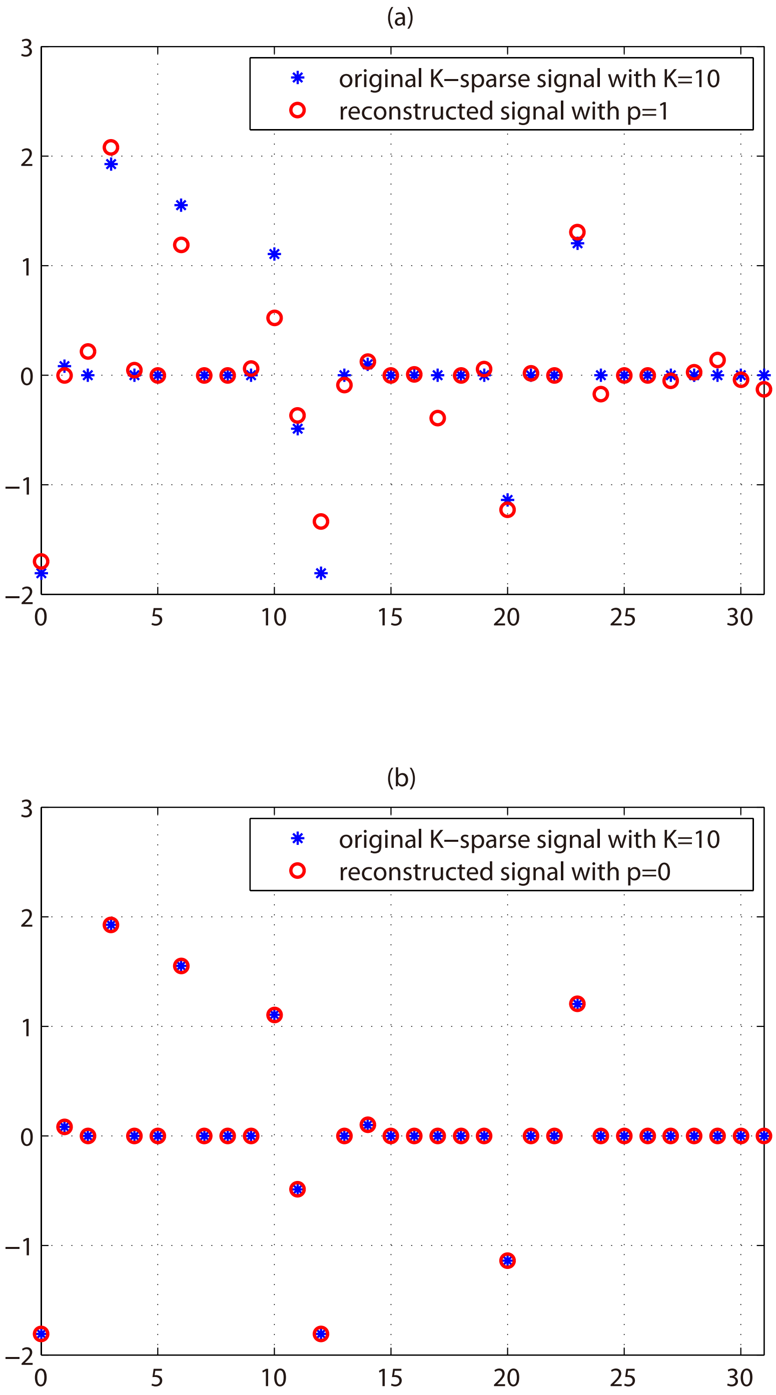


Figure 3.11: The original K-sparse signal versus the CS reconstructed signal with (a) $p = 1$; (b) $p = 0$.

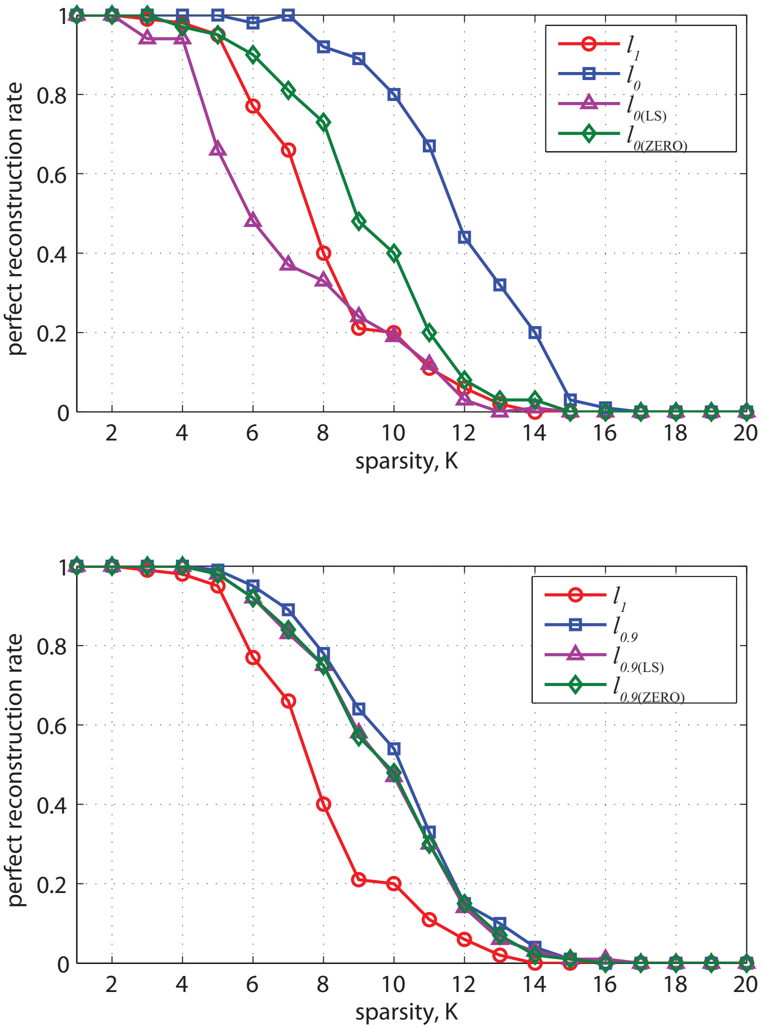
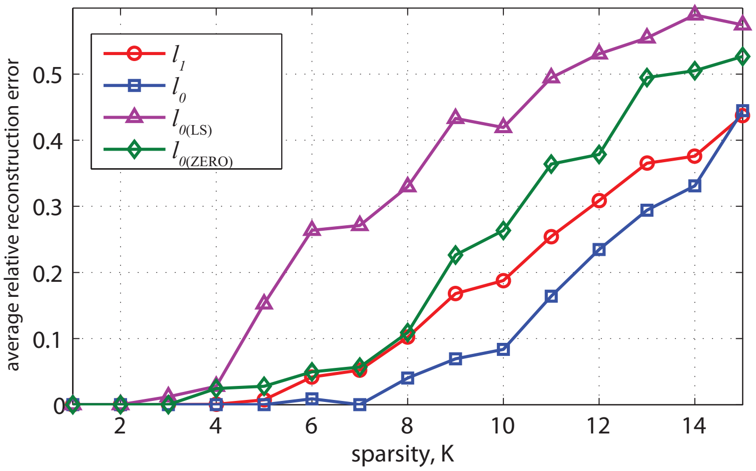


Figure 3.12: Rate of perfect reconstruction for ℓ_p-ℓ_2 problems for $p = 0$ and 0.9 obtained with different initial points over 100 runs with $N = 32$ and $M = 20$. The upper graph compares the ℓ_0 solution obtained by the proposed method with the ℓ_0 solution obtained by ℓ_p-MFISTA with the least-squares solution or the zero vector as the initial point. The lower graph does the comparison for the $p = 0.9$ counterpart. The curve corresponding to $p = 1$ is also shown as a comparison benchmark.



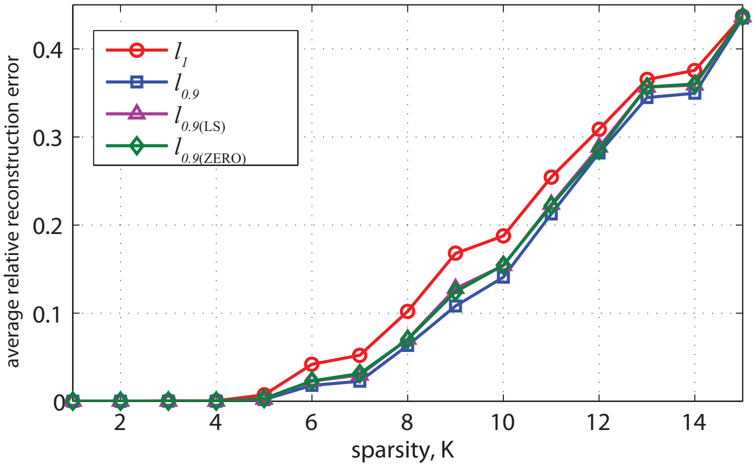


Figure 3.13: Average relative reconstruction errors for ℓ_p-ℓ_2 problems for $p = 0$ and 0.9 obtained with different initial points over 100 runs with $N = 32$ and $M = 20$. The upper graph compares the ℓ_0 solution obtained by the proposed method with the ℓ_0 solution obtained by ℓ_p-MFISTA with the least-squares solution or the zero vector as the initial point. The lower graph does the comparison for the $p = 0.9$ counterpart. The curve corresponding to $p = 1$ is also shown as a comparison benchmark.

3.2 Smoothed ℓ_p-ℓ_2 Solver for Signal Denoising

3.2.1 A Smoothed ℓ_p-ℓ_2 Solver and Its Fast Implementation

Although the ℓ_p-ℓ_2 problem (3.1) turns out to be a good model for signal denoising, there are issues to deal with. To start, we recall problem (3.1) given by

$$\text{minimize}\ \ F(\mathbf{s}) = \lambda||\mathbf{s}||_p^p + ||\mathbf{\Theta}\mathbf{s} - \mathbf{y}||^2$$

As demonstrated in Fig. 3.4, the denoising performance is significantly improved using a p less than 1. However, some ℓ_p SNR curves produced therein exhibit considerable oscillations with respect to a varying parameter λ, rendering the denoising performance unpredictable. The oscillations have to do with the fact that model (3.1) is designed for promoting sparsity, but not necessarily for higher SNR.

As a remedy to this problem, in this section we propose a smoothed ℓ_p-ℓ_2 solver that is able to produce stable recoveries for signal/image denoising. Let us first illustrate this concept by considering the case with $\mathbf{\Theta}$ orthogonal, which is initially analyzed in Sec. 3.1.2 where $F(\mathbf{s})$ in (3.1) is reduced to a simplified form in (3.7) as

$$F(\mathbf{s}) = \lambda\|\mathbf{s}\|_p^p + \|\mathbf{s} - \mathbf{c}\|^2.$$

Without loss of generality, we examine the single-variable function

$$u(s;\lambda) = \lambda|s|^p + (s-c)^2 \tag{3.9}$$

with $c > 0$ so that the absolute value sign of $|s|$ can be removed. By combining the graphs of λs^p and $(s-c)^2$, the presence of term λs^p yields a notch at $s = 0$ which is either a local or a global minimizer, depending on the value of λ. In effect, there is a value $\hat{\lambda} > 0$ at which the two minimizers are equal, hence both become global minimizers. The critical value $\hat{\lambda}$ and the locations of the two global minimizers, 0 and $\hat{s}$, can be determined by solving the following simultaneous equations.

$$\left.\frac{\partial u}{\partial s}\right|_{s=\hat{s},\lambda=\hat{\lambda}} = 0 \quad \text{and} \quad u(\hat{s};\hat{\lambda}) = u(0;\hat{\lambda})$$

In doing so, we obtain

$$\hat{s} = \frac{2(1-p)c}{2-p} \quad \text{and} \quad \hat{\lambda} = \frac{c^{2-p}}{1-p} \cdot \left[\frac{2(1-p)}{2-p}\right]^{2-p} \tag{3.10}$$

Note that $\hat{s}$ and $\hat{\lambda}$ computed from (3.10) satisfy $0 < \hat{s} < c$ and $\left.\frac{\partial^2 u}{\partial s^2}\right|_{s=\hat{s},\lambda=\hat{\lambda}} = 2-p > 0$, hence $\hat{s}$ is indeed a minimizer inside $[0, c]$.

On one hand, for a $\lambda < \hat{\lambda}$, the interior minimizer s^* determined by $\frac{\partial u}{\partial s} = 0$ and $\frac{\partial^2 u}{\partial s^2} > 0$ is the unique global minimizer of $u(s; \lambda)$; on the other hand for a $\lambda > \hat{\lambda}$, the origin $s^* = 0$ becomes the unique global minimizer. As a result, the global minimizer s^* jumps between the origin and the interior point $\hat{s}$ as λ varies across the critical value $\hat{\lambda}$ given by (3.10). Fig. 3.14 illustrates our analysis for the case of $p = 0.5$ and $c = 1$ in that (3.10) produces $\hat{\lambda} = 1.0887$. Fig. 3.14(a)-(c) show the global minimizers of $u(s; \lambda)$ for (a) $\lambda = 1.08 < \hat{\lambda}$, (b) $\lambda = \hat{\lambda}$, and (c) $\lambda = 1.09 > \hat{\lambda}$. The global minimizer of $u(s; \lambda)$, denoted by $s^*(\lambda)$, as a function of λ is depicted in Fig. 3.14(d) where its discontinuity at $\hat{\lambda} = 1.0887$ is evident.

The discontinuity of $s^*(\lambda)$ is undesirable as it degrades the stability and predictability of the solution from (3.7). As illustrated in Fig. 3.4, the SNR curves exhibit unpleasant jumps out of such discontinuity. As a remedy, below we propose a strategy that prefers a stable minimizer $\mathbf{s}^*$ rather than a global minimizer for (3.7) in case parameter λ is in a vicinity of the discontinuity point $\hat{\lambda}$. By assuming the value of λ falls within an interval $[\lambda_L, \lambda_H]$ and using (3.10) to evaluate the critical $\hat{\lambda}_i$ for each component c_i, each component s_i^* of the solution vector $\mathbf{s}^*$ is found as follows:

1. If $\hat{\lambda}_i \notin [\lambda_L, \lambda_H]$, solution jump will not occur, hence the global solution s_i^* can be found with stability: if $\hat{\lambda}_i < \lambda_L$, set $s_i^* = 0$; if $\hat{\lambda}_i > \lambda_H$, set s_i^* as the minimizer inside $[0, c_i]$ which can be efficiently identified using the ℓ_p-ℓ_2 solver `gsol` developed in Sec. 3.1.1.

2. If $\hat{\lambda}_i \in [\lambda_L, \lambda_H]$, to prevent solution jump, we take the unique global solution of $u(s)$ with $p = 1$ as s_i^*, which is simply the result of a soft-shrinkage operation as $s_i^* = \text{sign}(c_i) \cdot \max\{|c_i| - \lambda/2, 0\}$.

Although not a truly global solver, the solution procedure proposed above eliminates the jump phenomenon and offers a stable yet nearly global solution $\mathbf{s}^*$. With the same technique as that described in Sec. 3.1.1, the smoothed solver admits a

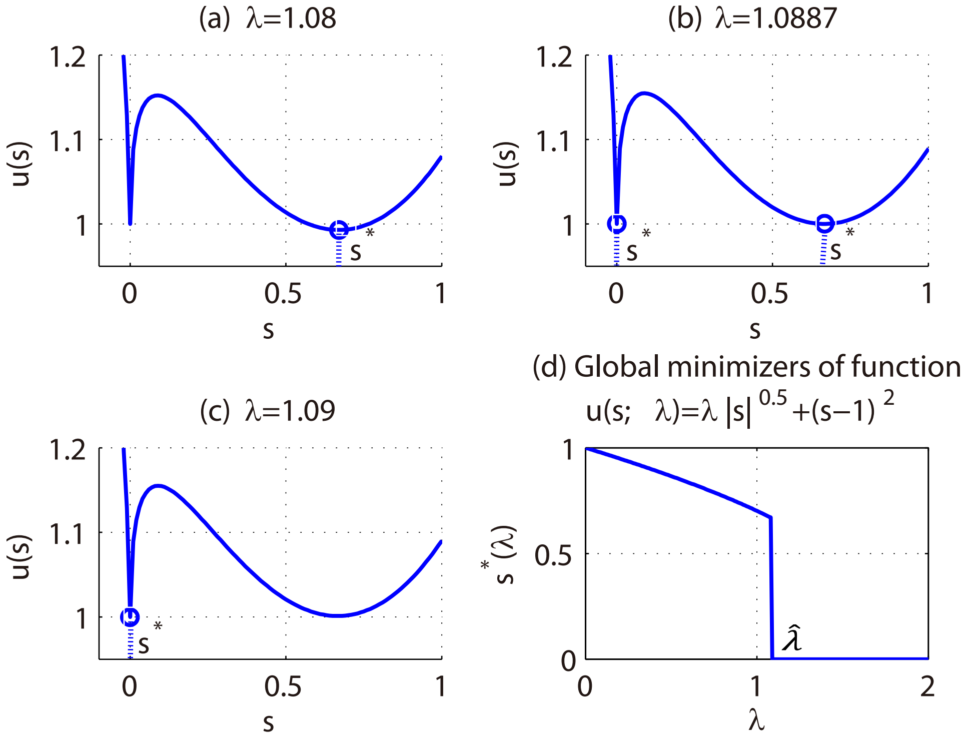


Figure 3.14: Global minimizer $s^*(\lambda)$ of $u(s;\lambda) = \lambda|s|^{0.5} + (s-1)^2$ (a) $\lambda = 1.08$, (b) $\lambda = \hat{\lambda} = 1.0887$, (c) $\lambda = 1.09$, (d) discontinuity of $s^*(\lambda)$ at $\hat{\lambda} = 1.0887$.

Algorithm 3.4	
Input Data	$\mathbf{c}$, λ, p, λ_L and λ_H.
Output Data	$\mathbf{s}^*$.
Step 1	Compute $\mathbf{c}_+ = \text{sign}(\mathbf{c}) \circ \mathbf{c}$.
Step 2	Compute $\hat{\boldsymbol{\lambda}} = \dfrac{\mathbf{c}_+.^{2-p}}{1-p} \cdot \left[\dfrac{2(1-p)}{2-p}\right]^{2-p}$.
Step 3	Define $\mathcal{J} = \{i : \hat{\lambda}_i \in [\lambda_L, \lambda_H]\}$ and $\mathcal{C} = \{i : \hat{\lambda}_i \notin [\lambda_L, \lambda_H]\}$. Define $\mathbf{c}_\mathcal{J} = \mathbf{c}(\mathcal{J})$ and $\mathbf{c}_\mathcal{C} = \mathbf{c}(\mathcal{C})$.
Step 4	Compute $\mathbf{s}_\mathcal{J} = \text{sign}(\mathbf{c}_\mathcal{J}) \circ \max(\lvert\mathbf{c}_\mathcal{J}\rvert - \lambda/2, 0)$ and $\mathbf{s}_\mathcal{C} = \text{argmin}\ \{\lambda\|\mathbf{s}\|_p^p + \|\mathbf{s} - \mathbf{c}_\mathcal{J}\|^2\}$.
Step 5	Set $\mathbf{s}^*(\mathcal{J}) = \mathbf{s}_\mathcal{J}$ and $\mathbf{s}^*(\mathcal{C}) = \mathbf{s}_\mathcal{C}$. Return $\mathbf{s}^*$.

Table 3.4: A smoothed ℓ_p-ℓ_2 solver for stable minimizer of (3.1) with orthogonal dictionary $\boldsymbol{\Theta}$

fast implementation which solves the N single-variable ℓ_p-ℓ_2 problems in parallel. A step-by-step description of a parallel implementation of the smoothed ℓ_p-ℓ_2 solver is described in Table 3.4 as Algorithm 3.4 where the data are processed in a vector-wise manner.

To investigate the circumstance that $\mathbf{\Theta} \in R^{M\times N}$ with $M < N$ is an overcomplete dictionary, we deal with the non-orthogonality of $\mathbf{\Theta}$ by an iterative technique that is in spirit similar to a proximal-point method employed in [8]: to update iterate $\mathbf{s}_k$ in the kth iteration to

$$\mathbf{s}_{k+1} = \underset{\mathbf{s}}{\operatorname{argmin}} \ \{\lambda\|\mathbf{s}\|_p^p + \frac{L}{2}\|\mathbf{s} - \mathbf{c}_k\|_2^2\} \tag{3.11}$$

where $\mathbf{c}_k = \mathbf{s}_k - \dfrac{2}{L}\mathbf{\Theta}^T(\mathbf{\Theta}\mathbf{s}_k - \mathbf{y})$ and L is the Lipschitz constant of the gradient of $\|\mathbf{\Theta}\mathbf{s} - \mathbf{y}\|_2^2$ given by $L = 2\lambda_{\max}(\mathbf{\Theta}^T\mathbf{\Theta})$. Note that for an orthogonal basis $\mathbf{\Theta}$, we have $L = 2$, $\mathbf{c}_k = \mathbf{\Theta}^T\mathbf{y} = \mathbf{c}$ and (3.11) becomes $\mathbf{s}^* = \operatorname{argmin}_{\mathbf{s}} \ \{\lambda\|\mathbf{s}\|_p^p + \|\mathbf{s} - \mathbf{c}\|_2^2\}$ which is exactly the case addressed in Sec. 3.2. Also note that the formulation differs from that of [8] as here we deal with a nonconvex objective function because $p \in (0, 1)$. The primary reason to employ (3.11) is that it is again a separable objective function whose solution was analyzed in detail in Sec. 3.2. Furthermore, formulation (3.11) allows us to incorporate FISTA [8] type of iteration into this formulation so as to accelerate the algorithm without substantial increase in computational complexity. Essentially a FISTA iteration modifies vector $\mathbf{c}_k$ to $\mathbf{c}_k = \mathbf{b}_k - \dfrac{2}{L}\mathbf{\Theta}^T(\mathbf{\Theta}\mathbf{b}_k - \mathbf{y})$ where $\mathbf{b}_k$ is updated using two previous iterates $\mathbf{s}_{k-1}$ and $\mathbf{s}_{k-2}$. More algorithmic details are provided in Table 3.5.

3.2.2 Performance of the Smoothed ℓ_p-ℓ_2 Solver in 1-D Signal Denoising with Orthogonal Dictionary

The smoothed ℓ_p-ℓ_2 solvers have so far been applied to denoising 1-D measurements. In this section, the results obtained from various simulation settings are compared with each other. The same noise-corrupted "`HeaviSine`" [50] signal as generated in Sec. 3.1.2 was adopted here for the simulation. Matrix $\mathbf{\Theta}$ was chosen as the same orthogonal 8-level Daubechies wavelet D8 basis. Standard deviation of the noise is $\sigma = 0.08$. The lower and upper bounds for λ were set to $\lambda_L = 0$ and $\lambda_H = 2\sigma\sqrt{2\lg N} \approx 0.35$, which corresponds to the universal soft shrinkage threshold [39]. With p fixed as one of the six values $\{1, 0.8, 0.6, 0.4, 0.2, 0\}$, Algorithm in Table

Algorithm 3.5	
Input	$\mathbf{y}$, $\mathbf{\Theta}$, λ, p, λ_L, λ_H and $\mathbf{s}_0$.
Output	$\mathbf{s}^*$.
Step 1	Compute the Lipschitz constant $L = 2\lambda_{\max}(\mathbf{\Theta}^T\mathbf{\Theta})$. Set the number of iterations K.
Step 2	Set $\mathbf{b}_1 = \mathbf{s}_0$, $t_1 = 1$ and $k = 1$.
Step 3	Compute $\mathbf{c}_k = \frac{2}{L}\mathbf{\Theta}^T(\mathbf{y} - \mathbf{\Theta}\mathbf{b}_k) + \mathbf{b}_k$, apply Algorithm 3.4 to solve $\mathbf{s}_k = \text{argmin}_{\mathbf{s}}\ \{\frac{2\lambda}{L}\|\mathbf{s}\|_p^p + \|\mathbf{s} - \mathbf{c}_k\|_2^2\}$ and compute $\begin{aligned} t_{k+1} &= \frac{1+\sqrt{1+4t_k^2}}{2} \\ \mathbf{b}_{k+1} &= \mathbf{s}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{s}_k - \mathbf{s}_{k-1}) \\ k &= k+1 \end{aligned}$
Step 4	If $k = K$, output $\mathbf{s}_k$ as solution $\mathbf{s}^*$ and terminate; otherwise repeat from Step 3.

Table 3.5: A smoothed ℓ_p-ℓ_2 solver for stable minimizer of (3.1) with overcomplete dictionary $\mathbf{\Theta}$

3.4 was applied to solve problem (3.1) with λ uniformly placed between λ_L and λ_H with a 0.01 interval.

The SNRs obtained versus λ for each p are depicted as six curves in Fig. 3.15. It is observed that for each fixed λ, using a $p < 1$ offers improved SNR relative to that obtained with $p = 1$ (BPDN); more importantly, for a fixed p the SNR is a smooth function of λ, and the value of λ achieving peak SNR gradually increases as p decreases. In comparison, we show in Fig. 3.16 the SNR profiles obtained by global solutions of (3.1). Most of the SNRs associated with $p < 1$ exhibit considerable oscillations – a sharp departure from the smooth concave SNR profiles as shown in Fig. 3.15.

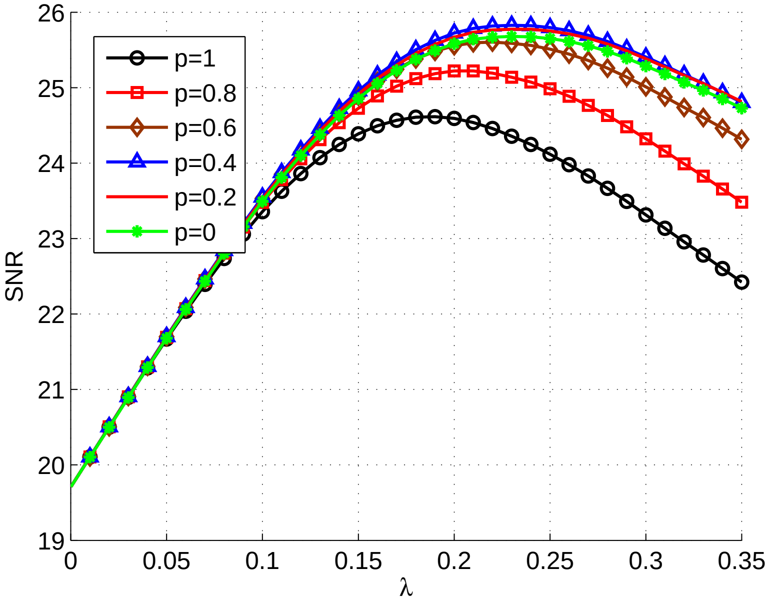


Figure 3.15: SNRs produced by denoising signal “`HeaviSine`” by Algorithm 3.4 with orthogonal $\mathbf{\Theta}$.

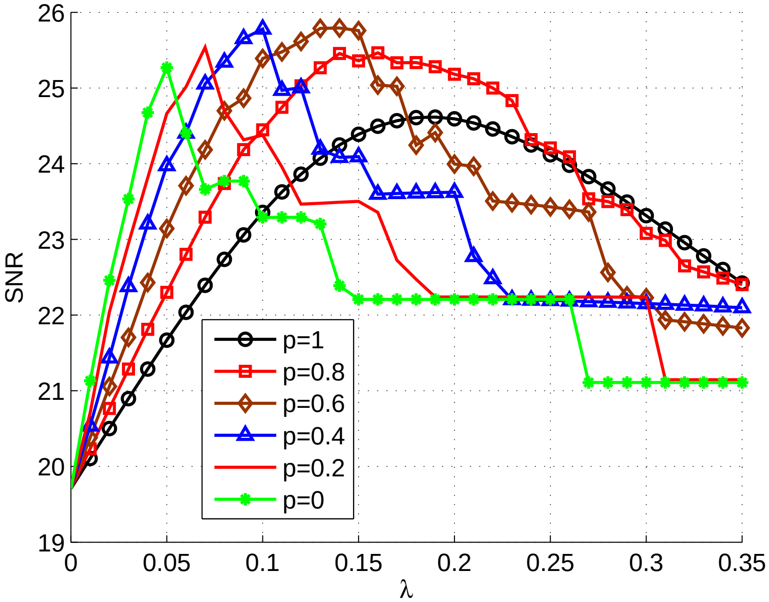


Figure 3.16: SNRs produced by denoising signal “`HeaviSine`” by global solution with orthogonal $\mathbf{\Theta}$.

3.2.3 Performance of the Smoothed ℓ_p-ℓ_2 Solver for Image Denoising

We end this chapter by demonstrating the effectiveness of our proposed technique for large-scale problems in image denoising. The simulation is carried on a number of images in MATLAB. For experimental purposes, each 256×256 clean image (normalized with gray level between 0 and 1) was corrupted with Gaussian noise with zero mean and standard deviation $\sigma = 0.08, 0.10$ and 0.12 separately. An 8-level 2-D Daubechies D8 wavelet was used as the sparsifying basis. The lower and upper bounds for λ were set to $\lambda_L = 0$ and $\lambda_H = 2\sigma\sqrt{2 \lg 256^2}$, which corresponds to the universal soft shrinkage threshold [39]. The universal threshold is the optimal threshold in the asymptotic sense. In practice, the best empirical thresholds are much lower than this value, independent of the wavelet used. It therefore seems that the universal threshold is not useful to determine a threshold, but useful for obtain a starting value when nothing is known of the signal condition. Fast 2-D wavelet transform is used to avoid the large-scale matrix-vector multiplication.

To restore the images, the global ℓ_p-ℓ_2 solver in Algorithm 3.1 and the smoothed ℓ_p-ℓ_2 solver in Algorithm 3.4 were respectively applied to problem (3.1) with the threshold chosen as $\lambda = \lambda_H/2$. For different σ, SNRs (in dB) of the noisy image, the denoised image by basis pursuit ($p = 1$), the denoised image by global solver with $p = 0$ and the denoised image by smoothed solver with $p = 0$ are listed in Tables 3.6 and 3.7. By observing the data, we see that the image denoised by the global ℓ_p-ℓ_2 method with $p = 0$ has the smallest SNR. On the other hand, the smoothed ℓ_p-ℓ_2 method with $p = 0$ significantly outperforms the global one in terms of the SNR. It is further found that the SNR obtained through the smoothed method is consistently higher than that obtained by the BP benchmark for different images and noise levels.

For better visual observations, images of "`zelda`", "`reschart`", "`lena`" and "`circles`" denoised using different algorithms are illustrated in Fig. 3.17, where each row from left to right is composed of the noisy image, the BP denoised image, the denoised image by the global ℓ_p-ℓ_2 solver with $p = 0$ and the denoised image by the smoothed ℓ_p-ℓ_2 solver with $p = 0$. From Fig. 3.17, we see that the ℓ_0 global solution corresponds to the worst image quality. On the other hand, by using Algorithm 3.4 to reach a smoothed solution for the $p = 0$ case, quality of the denoised image is superb. The observation is indeed in agreement with the SNRs given in Tables 3.6 and 3.7.

Original Images	Noisy Images	BP	Global ℓ_0	Smoothed ℓ_0
`cameraman`	18.8546	22.2438	18.2188	22.4178
`zelda`	14.8966	19.6889	16.0739	19.8399
`shuttle`	16.3139	17.3914	12.9512	17.4999
`reschart`	23.8479	26.5138	21.9062	26.9276
`crosses`	9.2532	13.0662	7.7553	13.3254
`man`	16.8351	17.8921	13.8777	17.9965
`fruits`	19.6682	24.7864	20.9545	25.0135
`lena`	17.2656	21.0194	17.1351	21.2199
`circles`	18.4722	24.0046	19.7252	24.3100

Table 3.6: Comparison between basis pursuit, global ℓ_p-ℓ_2 solver and smoothed ℓ_p-ℓ_2 solver for denoising images corrupted by Gaussian noise with standard deviation $\sigma = 0.06$

Original Images	Noisy Images	BP	Global ℓ_0	Smoothed ℓ_0
`cameraman`	16.3558	20.7632	17.5960	20.9249
`zelda`	12.3978	18.2712	15.4430	18.4179
`shuttle`	13.8151	15.9141	12.4373	16.0195
`reschart`	21.3491	24.5732	20.8815	24.9332
`crosses`	6.7545	11.1173	6.8332	11.2881
`man`	14.3364	16.5181	13.3481	16.6211
`fruits`	17.1694	23.2176	20.1135	23.4441
`lena`	14.7668	19.5276	16.5046	19.7487
`circles`	15.9734	22.1335	18.7916	22.4024

Table 3.7: Comparison between basis pursuit, global ℓ_p-ℓ_2 solver and smoothed ℓ_p-ℓ_2 solver for denoising images corrupted by Gaussian noise with standard deviation $\sigma = 0.08$

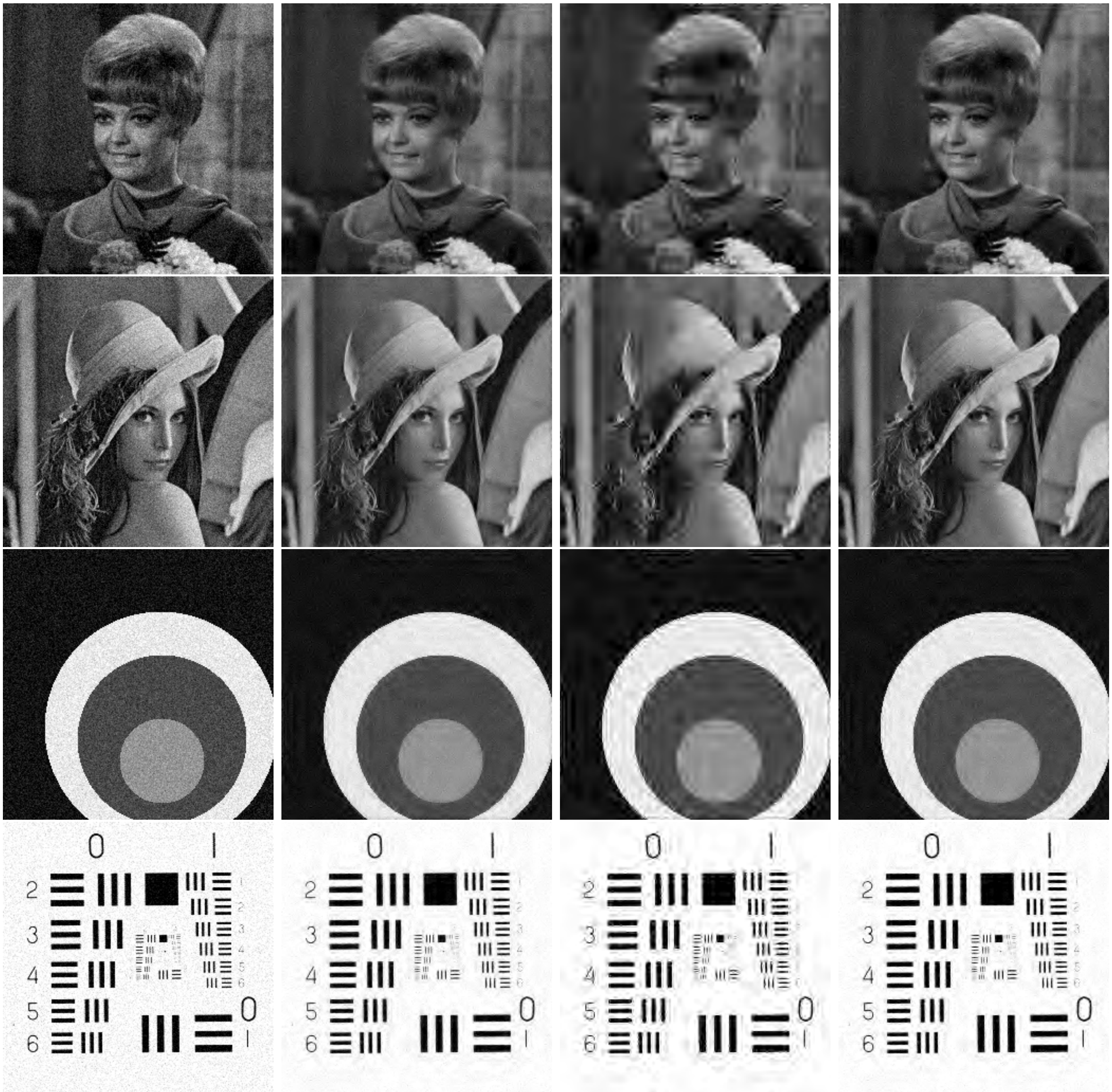


Figure 3.17: Denoised images for "`zelda`", "`lena`", "`circles`" and "`reschart`". Standard deviation of the Gaussian noise is 0.06.

3.2.4 Performance of the Smoothed ℓ_p-ℓ_2 Solver in 1-D Signal Denoising with Overcomplete Dictionary

In this part of the simulations, we investigate denoising of 1-D noisy signal with overcomplete dictionary. The "`HeaviSine`" signal $\mathbf{x}$ of length $N = 256$ was corrupted with additive Gaussian white Gaussian noise $\mathbf{n}$ with zero mean and standard deviation $\sigma = 0.3$.

A dictionary $\mathbf{\Theta} = [\mathbf{\Theta}_1 \ \mathbf{\Theta}_2]$ of size 256×512 with $\mathbf{\Theta}_1$ the 8-level Daubechies D8 wavelet basis and $\mathbf{\Theta}_2$ the 1-level Haar wavelet basis was used as an overcomplete dictionary. The lower and upper bounds of λ were set to $\lambda_L = 0$ and $\lambda_H = 1.4$. Because both $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ are orthogonal, the Lipschitz constant $L = 2\lambda_{\max}(\mathbf{\Theta}\mathbf{\Theta}^T) = 4$. Algorithm in Table 3.5 was applied to each of the six cases of $p \in \{1, 0.8, 0.6, 0.4, 0.2, 0\}$, where problem (3.1) was solved for each of 141 λ's that were equally placed over $[0, 1.4]$. In our implementation, the solution $\mathbf{s}(\lambda)$ obtained from a given λ was used as the initial point for the algorithm to proceed with the subsequent value of λ. The use of this better initial point was found helpful in reducing the number of iterations required. The SNRs obtained are shown in Fig. 3.18.

We see that the observations made in Sec. 3.2.2 for the case of orthogonal basis also hold here, except that the best performance in the present case was achieved with $p = 0.4$ at $\lambda = 1.17$, offering an SNR of 27.12 dB which is 0.9 dB higher than the maximum SNR obtained by Algorithm 3.4 for the orthogonal basis. For comparison, Fig. 3.19 depicts the SNRs obtained by replacing the 2nd sub-step in Step 3 of Algorithm 3.5 with $\mathbf{s}_k$ = global minimizer of $\{\frac{2\lambda}{L}||\mathbf{s}||_p^p + ||\mathbf{s} - \mathbf{c}_k||_2^2\}$. Like the case when $\mathbf{\Theta}$ is orthogonal, the SNRs with $p < 1$ show a great deal of instability with respect to λ.

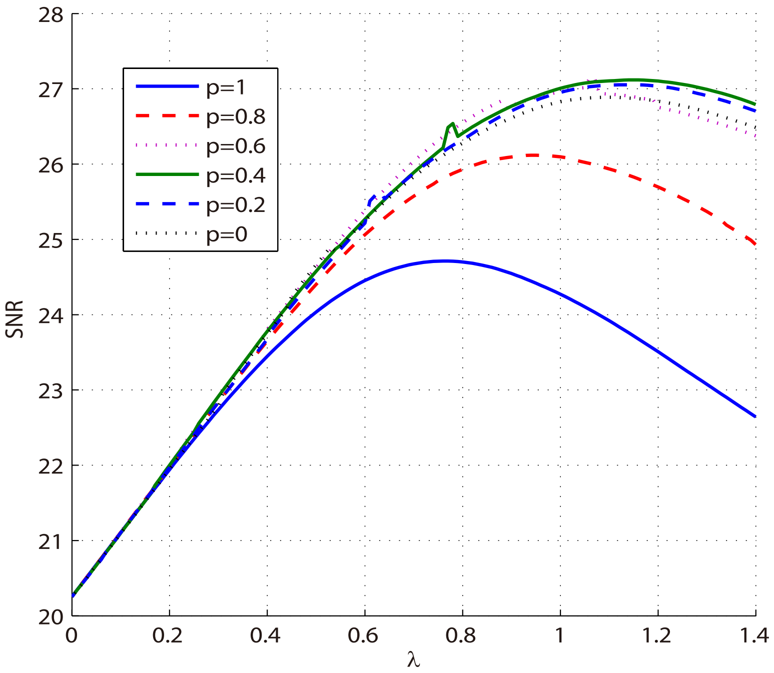


Figure 3.18: SNRs produced by denoising signal “HeaviSine” by Algorithm 3.5 with overcomplete $\mathbf{\Theta}$.

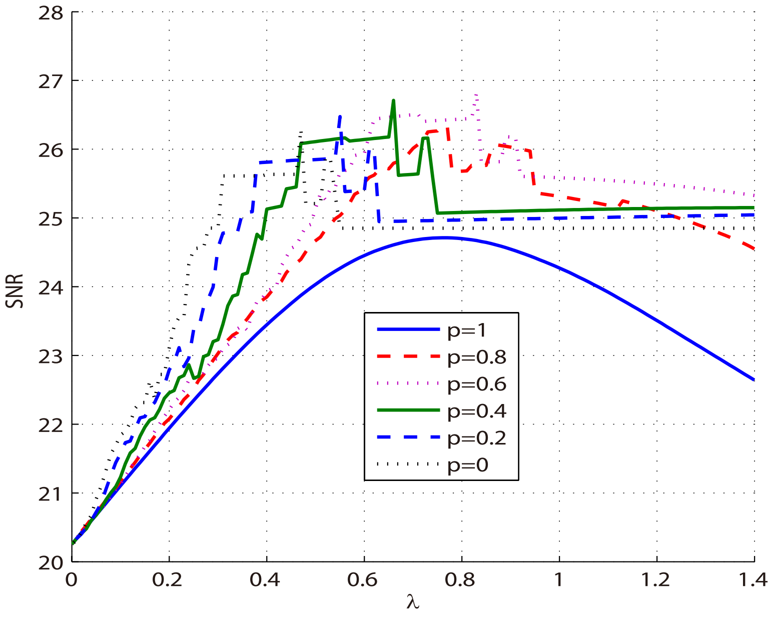


Figure 3.19: SNRs produced by denoising signal “HeaviSine” by replacing the 2nd sub-step in Step 3 of Algorithm 3.5 by $\mathbf{s}_k$ = global minimizer of $\{\frac{2\lambda}{L}\|\mathbf{s}\|_p^p + ||\mathbf{s}-\mathbf{c}_k||_2^2\}$ with overcomplete $\mathbf{\Theta}$.

Chapter 4

Fast Dual-Based Linearized Bregman Algorithm for Compressive Sensing

A central problem in compressive sensing is the recovery of a sparse signal using a relatively small number of linear measurements. The basis pursuit (BP) [39] has been a successful formulation for this signal reconstruction problem. Among other things, linearized Bregman (LB) [90, 119, 120] methods proposed recently are found effective to solve BP. In this Chapter, we examine the equality constrained problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & J(\mathbf{x}) && \text{(4.1a)} \\ \text{subject to:} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} && \text{(4.1b)} \end{aligned}$$

where $J(\mathbf{x})$ is a continuous (but non-differentiable) objective function. As introduced in Sec. 2.5, linearized Bregman (LB) algorithms based on Bregman distance [12] are efficient techniques for above-mentioned problems, especially when the problem considered is of large scale.

The chapter is organized as follows. The dual problem and a dual-based linearized Bregman method are first discussed in Sec. 4.1 and 4.2, respectively. A fast linearized Bregman algorithm applied to the dual formulation that accelerates the conventional LB iterations is proposed in Sec. 4.3. Finally, the performance of the proposed algorithm is evaluated and compared with the conventional LB algorithm in compressive sampling of 1-D sparse signals and digital images.

4.1 Lagrangian Dual of Problem (2.15)

Recently, the LB method for problem (4.1) is shown to be equivalent to a gradient descent algorithm applied to the Lagrangian dual of (2.15) [119]. The Lagrangian dual of problem (2.15) assumes the form

$$\max_{\mathbf{y}} \min_{\mathbf{x}} \; J(\mathbf{x}) + \frac{1}{2\mu}\|\mathbf{x}\|^2 - \langle \mathbf{y}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle \tag{4.2}$$

The dual function [4, 10] is

$$D(\mathbf{y}) = \inf_{\mathbf{x}} \; \{J(\mathbf{x}) + \frac{1}{2\mu}\|\mathbf{x}\|^2 - \langle \mathbf{y}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle\} \tag{4.3}$$

where $\mathbf{y}$ is the dual variable or Lagrange multiplier. The dual problem equivalent to (4.2) is expressed as

$$\underset{\mathbf{y}}{\text{minimize}} \; E(\mathbf{y}) \tag{4.4}$$

where $E(\mathbf{y}) = -D(\mathbf{y})$.

The gradient $\nabla E(\mathbf{y})$ can be evaluated assuming that $E(\mathbf{y})$ is differentiable. It is first defined that

$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \; \{J(\mathbf{x}) + \frac{1}{2\mu}\|\mathbf{x}\|^2 - \langle \mathbf{y}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle\} \tag{4.5}$$

then $\nabla E(\mathbf{y}) = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}$ which corresponds to the residual for the equality constraint (see [10] for background). It is known that $E(\mathbf{y})$ is continuously differentiable. If $J(\cdot) = \|\cdot\|_1$, then ∇E is Lipschitz continuous with the smallest Lipschitz constant $L = \mu\|\mathbf{A}\mathbf{A}^T\|$. Consequently, the dual problem can be solved by means of gradient-based techniques such as limited-memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [74], conjugate gradient [99], and Nesterov's methods [86], possibly in conjunction with efficient line search (e.g., Barzilai-Borwein [42, 56]) techniques.

4.2 A Dual-Based Linearized Bregman Method

The ℓ_1-norm related optimization is an essential component in compressive sensing applications. In this section, we focus on the ℓ_1 case, i.e., $J(\mathbf{x}) = \|\mathbf{x}\|_1$. Because $E(\mathbf{y})$

is convex with Lipschitz continuous $\nabla E(\mathbf{y})$, it follows that

$$E(\mathbf{y}) \leq E(\mathbf{y}^k) + \langle \mathbf{y} - \mathbf{y}^k, \nabla E(\mathbf{y}^k) \rangle + \frac{L}{2} \|\mathbf{y} - \mathbf{y}^k\|^2 \tag{4.6}$$

for any $\mathbf{y}$ and $\mathbf{y}^k$. In a steepest descent method [4], iterate $\mathbf{y}^k$ is updated to $\mathbf{y}^{k+1}$ with

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \alpha_k \nabla E(\mathbf{y}^k) \tag{4.7}$$

where $\alpha_k > 0$ is a scalar step size. Note that iterate $\mathbf{y}^{k+1}$ may be interpreted as the solution to a quadratic problem

$$\mathbf{y}^{k+1} = \operatorname*{argmin}_{\mathbf{y}} H(\mathbf{y}, \mathbf{y}^k)$$

where

$$H(\mathbf{y}, \mathbf{y}^k) = E(\mathbf{y}^k) + \langle \mathbf{y} - \mathbf{y}^k, \nabla E(\mathbf{y}^k) \rangle + \frac{1}{2\alpha_k} \|\mathbf{y} - \mathbf{y}^k\|^2.$$

By comparing the equation above with (4.6), we see that the quadratic function $H(\mathbf{y}, \mathbf{y}^k)$ serves as a reasonable approximation of $E(\mathbf{y})$ at $\mathbf{y} = \mathbf{y}^k$ if α_k is set to $1/L$ where $L = \mu\|\mathbf{A}\mathbf{A}^T\|$. Thus, at the $(k+1)$th iteration, we compute

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \frac{1}{L} \nabla E(\mathbf{y}^k) = \mathbf{y}^k - \frac{1}{L}(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \tag{4.8}$$

where $\mathbf{x}^{k+1}$ is computed by

$$\begin{aligned} \mathbf{x}^{k+1} &= \operatorname*{argmin}_{\mathbf{x}} \{ \|\mathbf{x}\|_1 + \frac{1}{2\mu} \|\mathbf{x}\|^2 - \langle \mathbf{y}^k, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle \} \\ &= \operatorname*{argmin}_{\mathbf{x}} \{ \|\mathbf{x}\|_1 + \frac{1}{2\mu} \|\mathbf{x} - \mu \mathbf{A}^T \mathbf{y}^k\|^2 \}. \end{aligned}$$

By defining $\mathcal{T}_\alpha : \mathbb{R}^N \to \mathbb{R}^N$ as the soft-shrinkage operator, i.e., $\mathcal{T}_\alpha(\mathbf{z}) = \operatorname{sgn}(\mathbf{z}) \circ \max\{|\mathbf{z}| - \alpha, 0\}$, we have

$$\mathbf{x}^{k+1} = \mathcal{T}_\mu(\mu \mathbf{A}^T \mathbf{y}^k) = \mu \mathcal{T}_1(\mathbf{A}^T \mathbf{y}^k). \tag{4.9}$$

The above algorithm is described below as Algorithm 4.1, where the initial iterate has been specified as $\mathbf{y}^0 = \frac{1}{L}\mathbf{b}$. Note that the iteration (4.9) corresponds to the conventional gradient-descent method, or the iterative shrinkage-thresholding algorithm (see Sec. 2.3), and is known to possess a worst-case convergence rate of $O(1/k)$ where

k refers to the number of iterations.

Algorithm 4.1 Dual-Based LB

```
Input: $\mu > 0$, $\mathbf{A}$, $\mathbf{b}$, $L = \mu\|\mathbf{A}\mathbf{A}^T\|$ and $\mathbf{y}^0 = \frac{1}{L}\mathbf{b}$.
for $k = 0, 1, ...$ do
    $\mathbf{x}^{k+1} = \mu\mathcal{T}_1(\mathbf{A}^T\mathbf{y}^k)$;
    $\mathbf{y}^{k+1} = \mathbf{y}^k - \frac{1}{L}(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})$;
end for
```

It has been established in [69, 119] that the dual-based LB method (Algorithm 4.1) and the conventional LB in Algorithm 2.4 are equivalent. In the following, we provide a convergence proof for Algorithm 4.1. Since $E(\mathbf{y})$ is convex, it follows that

$$E(\mathbf{y}) \geq E(\mathbf{y}^k) + \langle \mathbf{y} - \mathbf{y}^k, \nabla E(\mathbf{y}^k) \rangle. \tag{4.10}$$

The above inequality together with (4.6) when $\mathbf{y} = \mathbf{y}^{k+1}$ produces

$$\begin{aligned} & E(\mathbf{y}) - E(\mathbf{y}^{k+1}) \\ & \geq \langle \mathbf{y} - \mathbf{y}^{k+1}, \nabla E(\mathbf{y}^k) \rangle - \frac{L}{2}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 \\ & = \frac{L}{2}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 + L\langle \mathbf{y} - \mathbf{y}^k, \mathbf{y}^k - \mathbf{y}^{k+1} \rangle. \end{aligned} \tag{4.11}$$

In particular, by substituting $\mathbf{y} = \mathbf{y}^*$ (the global minimizer of $E(\mathbf{y})$) and $\mathbf{y} = \mathbf{y}^k$ in (4.11) respectively, the following two inequalities hold,

$$E(\mathbf{y}^*) - E(\mathbf{y}^{k+1}) \geq \frac{L}{2}(\|\mathbf{y}^* - \mathbf{y}^{k+1}\|^2 - \|\mathbf{y}^* - \mathbf{y}^k\|^2), \tag{4.12}$$

and

$$E(\mathbf{y}^k) - E(\mathbf{y}^{k+1}) \geq \frac{L}{2}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2. \tag{4.13}$$

Summing inequality (4.12) over $k = 0, \cdots, K-1$ produces

$$KE(\mathbf{y}^*) - \sum_{k=0}^{K-1} E(\mathbf{y}^{k+1}) \geq \frac{L}{2}(\|\mathbf{y}^* - \mathbf{y}^K\|^2 - \|\mathbf{y}^* - \mathbf{y}^0\|^2). \tag{4.14}$$

In a similar way, we multiply (4.13) by k, then sum the terms over $k = 0, \cdots, K-1$

to obtain

$$\sum_{k=0}^{K-1} k(E(\mathbf{y}^k) - E(\mathbf{y}^{k+1})) \geq \frac{L}{2}\sum_{k=0}^{K-1} k\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2. \tag{4.15}$$

The term on the left-hand side of (4.15) gives

$$\begin{aligned}
&\sum_{k=0}^{K-1} k(E(\mathbf{y}^k) - E(\mathbf{y}^{k+1})) \\
&\quad = \sum_{k=0}^{K-1} \big(kE(\mathbf{y}^k) - (k+1)E(\mathbf{y}^{k+1}) + E(\mathbf{y}^{k+1})\big) \\
&\quad = -KE(\mathbf{y}^K) + \sum_{k=0}^{K-1} E(\mathbf{y}^{k+1}),
\end{aligned}$$

hence we have

$$-KE(\mathbf{y}^K) + \sum_{k=0}^{K-1} E(\mathbf{y}^{k+1}) \geq \frac{L}{2}\sum_{k=0}^{K-1} k\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2. \tag{4.16}$$

Adding (4.16) to (4.14) yields

$$KE(\mathbf{y}^*) - KE(\mathbf{y}^K) \geq -\frac{L}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2 + c$$

where $c \geq 0$. Therefore,

$$E(\mathbf{y}^k) - E(\mathbf{y}^*) \leq \frac{L\|\mathbf{y}^0 - \mathbf{y}^*\|^2}{2k} \tag{4.17}$$

The estimate in (4.17) indicates that Algorithm 4.1 shares a convergence rate of $O(1/k)$. That is, $\mathbf{y}^k$ is an ε-optimal solution if $k \geq \lceil C/\varepsilon \rceil$ with $C = L\|\mathbf{y}^0 - \mathbf{y}^*\|^2/2$. When $\{\mathbf{y}^k\}$ converges to $\mathbf{y}^*$, with the same rate the sequence $\{\mathbf{x}^k\}$ converges to $\mathbf{x}_\mu$, the unique minimizer of (2.15). In addition, we observe that the sequence of function values $\{E(\mathbf{y}^k)\}$ produced by Algorithm 4.1 is non-increasing, as shown by (4.13). Furthermore, if we define the Lagrangian function for (2.15) as

$$\mathcal{L}_\mu(\mathbf{x}, \mathbf{y}) = \|\mathbf{x}\|_1 + \frac{1}{2\mu}\|\mathbf{x}\|^2 - \langle \mathbf{y}, \mathbf{A}\mathbf{x} - \mathbf{b}\rangle, \tag{4.18}$$

then Algorithm 4.1 implies that

$$E(\mathbf{y}^k) = -\mathcal{L}_\mu(\mathbf{x}^{k+1}, \mathbf{y}^k), \tag{4.19a}$$

$$E(\mathbf{y}^*) = -\mathcal{L}_\mu(\mathbf{x}_\mu, \mathbf{y}^*). \tag{4.19b}$$

Hence

$$\mathcal{L}_\mu(\mathbf{x}_\mu, \mathbf{y}^*) - \mathcal{L}_\mu(\mathbf{x}^{k+1}, \mathbf{y}^k) \leq \frac{L\|\mathbf{y}^0 - \mathbf{y}^*\|^2}{2k}. \tag{4.20}$$

Thus, $(\mathbf{x}^{k+1}, \mathbf{y}^k)$ is an ε-optimal solution to problem (2.15) with respect to the Lagrangian function if $k \geq \lceil C/\epsilon \rceil$ with $C = L\|\mathbf{y}^0 - \mathbf{y}^*\|^2/2$.

4.3 A Fast Dual-Based Linearized Bregman Method

In [119], Yin considered several techniques such as Barzilai-Borwein line search and limited memory BFGS (L-BFGS) to accelerate the classical gradient descent method. In addition, a very recent manuscript [69] deals with the CS problem in the dual space by utilizing the acceleration technique proposed by Nesterov [86]. On the other hand, Beck and Teboulle devise a faster method called FISTA [8]. While both FISTA and Nesterov's method are proven to converge with the same rate, the two schemes are remarkably different both conceptually and computationally [8]. Since FISTA is a proximal subgradient algorithm, it is simpler than Nesterov's method from an implementation perspective.

Inspired by Beck and Teboulle [8], we propose a fast iteration scheme by carrying out FISTA type of iterations in the dual space. Specifically, we perform gradient projection with a new iterate $\mathbf{z}^{k+1}$ as

$$\mathbf{y}^{k+1} = \mathbf{z}^{k+1} - \frac{1}{L}\nabla E(\mathbf{z}^{k+1}) \tag{4.21}$$

where

$$\mathbf{z}^{k+1} = \mathbf{y}^k + \frac{t_k - 1}{t_{k+1}}(\mathbf{y}^k - \mathbf{y}^{k-1}). \tag{4.22}$$

and the balancing parameter t_k has an iterative formula

$$t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2 \tag{4.23}$$

starting from the initial $t_0 = 0$. The main difference between (4.21) and (4.7) is that the current iteration is not involved in the point $\mathbf{y}^k$, but rather in point $\mathbf{z}^{k+1}$ which uses a very specific linear combination of two preceding points $\{\mathbf{y}^k, \mathbf{y}^{k-1}\}$. Obviously the additional computation required for the fast algorithm is insignificant. The new iteration however possesses a faster convergence rate of $O(1/k^2)$ as opposed to the conventional $O(1/k)$. We remark that the specific formula of the linear combination (4.22) and the computation of parameter t_k in (4.23) are the same as in FISTA [8].

With $\nabla E(\mathbf{y}) = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}$, we therefore set $\nabla E(\mathbf{z}^{k+1})$ in (4.21) to

$$\nabla E(\mathbf{z}^{k+1}) = \mathbf{A}\mathbf{x}^{k+1} - \mathbf{b} \tag{4.24}$$

where, as suggested by (4.5), $\mathbf{x}^{k+1}$ is obtained by

$$\begin{aligned}\mathbf{x}^{k+1} &= \underset{\mathbf{x}}{\operatorname{argmin}}\{\|\mathbf{x}\|_1 + \frac{1}{2\mu}\|\mathbf{x}\|^2 - \langle \mathbf{z}^{k+1}, \mathbf{A}\mathbf{x} - \mathbf{b}\rangle\} \\ &= \mathcal{T}_\mu(\mu\mathbf{A}^T\mathbf{z}^{k+1}) = \mu\mathcal{T}_1(\mathbf{A}^T\mathbf{z}^{k+1})\end{aligned}$$

Summarizing the iteration procedures described above, we have the fast dual-based linearized Bregman algorithm as Algorithm 4.2.

Algorithm 4.2 Fast Dual-Based LB

```
Input: $\mu > 0$, $\mathbf{A}$, $\mathbf{b}$, $L = \mu\|\mathbf{A}\mathbf{A}^T\|$, $\mathbf{y}^{-1} = \mathbf{y}^0 = \frac{1}{L}\mathbf{b}$ and $t_0 = 1$.
for $k = 0, 1, ..., K$ do
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
    $\mathbf{z}^{k+1} = \mathbf{y}^k + \frac{t_k - 1}{t_{k+1}}(\mathbf{y}^k - \mathbf{y}^{k-1})$;
    $\mathbf{x}^{k+1} = \mu\mathcal{T}_1(\mathbf{A}^T\mathbf{z}^{k+1})$;
    $\mathbf{y}^{k+1} = \mathbf{z}^{k+1} - \frac{1}{L}(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})$;
end for
```

In the following, we sketch a proof to show that Algorithm 4.2. has a convergence rate of $O(1/k^2)$. The proof is based on the fact [8] that if $\{a_k, b_k\}$ are positive sequences of reals satisfying

$$a_k - a_{k+1} \geq b_{k+1} - b_k \text{ and } a_1 + b_1 \leq c \tag{4.25}$$

for some $c > 0$ and $k \geq 1$, then $a_k < c$.

Let $\{\mathbf{y}^k\}$ be a sequence generated by Algorithm 4.2, $a_k = 2t_k^2 v_k/L$, $b_k = \|\mathbf{u}_k\|^2$, $c = \|\mathbf{y}^0 - \mathbf{y}^*\|^2$ with $v_k = E(\mathbf{y}^k) - E(\mathbf{y}^*)$ and $\mathbf{u}_k = t_k\mathbf{y}^k - (t_k - 1)\mathbf{y}^{k-1} - \mathbf{y}^*$. It can be shown that with the above assignments (4.25) is satisfied (readers are referred to [8] for more details). Hence $a_k < c$ which implies that

$$E(\mathbf{y}^k) - E(\mathbf{y}^*) < \frac{L(\|\mathbf{y}^0 - \mathbf{y}^*\|^2)}{2t_k^2}. \tag{4.26}$$

It can also be verified that the sequence t_k produced by Algorithm 4.2 satisfies $t_k \geq (k+1)/2$, which in conjunction with (4.26) shows that for $k \geq 1$

$$E(\mathbf{y}^k) - E(\mathbf{y}^*) < \frac{2L\|\mathbf{y}^0 - \mathbf{y}^*\|^2}{(k+1)^2}. \tag{4.27}$$

Hence $\mathbf{y}^k$ is an ϵ-optimal solution with respect to the dual function $E(\mathbf{y})$ if $k > \lceil C/\sqrt{\varepsilon} - 1 \rceil$ where $C = \sqrt{2L}\|\mathbf{y}^0 - \mathbf{y}^*\|$. As $\{\mathbf{y}^k\}$ converges to $\mathbf{y}^*$, the sequence $\{\mathbf{x}^k\}$ converges to $\mathbf{x}_\mu$, the unique minimizer of (2.15) with the same rate of $O(1/k^2)$.

We remark that Algorithm 4.2 has the advantage over FISTA in the sense that FISTA is only limited to minimizing the unconstrained ℓ_1-ℓ_2 problem [115], while (4.1) models a broader class of problems. In addition, it typically takes a large number of iterations for FISTA to converge to a solution that satisfies equality constraints $\mathbf{Ax} = \mathbf{b}$. Unlike FISTA, the proposed Algorithm 4.2 is associated with a dual problem of (2.15). As a result, the method is able to efficiently deal with equality constrained CS problem with fast convergence.

4.4 Performance Evaluation of Fast Dual-Based Linearized Bregman Method

4.4.1 Compressive Sensing of 1-D Signals

In the first set of examples, a partial discrete cosine transform (DCT) matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ was used as the measurement matrix whose M rows were chosen randomly from an $N \times N$ DCT matrix with $N = 4 \times 10^3$, 2×10^4 and 5×10^4 respectively, and $M = 0.5N$. In each case, a K-sparse test signal $\mathbf{x}^* \in \mathbb{R}^N$, with $K = 0.05N$ and $0.02N$ respectively, was constructed by assigning K values that are randomly drawn from $\mathcal{U}(-1, 1)$ (i.e., `2*rand(K,1)-1`) to K randomly selected locations in an otherwise zero vector of length N. We remark that partial DCT matrix is known to be efficient for compressive sensing, and both $\mathbf{Ax}$ and $\mathbf{A}^T\mathbf{u}$ can be carried out efficiently by fast DCT or the inverse DCT. The observed data $\mathbf{b}$ was set to $\mathbf{b} = \mathbf{Ax}^*$.

Algorithm 4.2 was implemented and compared with the conventional LB method [120]. The measurement matrix constructed above implies that $L = \mu$ where μ was set to 10 in the simulation. The algorithms were terminated when $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|/\|\mathbf{b}\| < 10^{-5}$ or the number of iterations exceeds 10^4. The performance of the algorithms was measured in terms of number of iterations (NoI) and CPU time using a PC laptop with a 2.67 GHz Intel quad-core processor. The results are summarized in Tables 4.1 and 4.2, where the reconstructed signal is denoted as $\mathbf{x}_p$, which clearly indicate improved performance offered by Algorithm 4.2 relative to the conventional LB method.

N	M	$\|\mathbf{x}^*\|_0$	NoI	$\frac{\|\mathbf{x}_p - \mathbf{x}^*\|}{\|\mathbf{x}^*\|}$	time (s)
4000	2000	$0.05N$	4011	1.0355e-5	10.7
20000	10000		10000+	N/A	90.2+
50000	25000		10000+	N/A	238.4+
4000	1000	$0.02N$	7096	1.1380e-5	17.3
20000	5000		10000+	N/A	84.6+
50000	12500		10000+	N/A	223.1+

Table 4.1: Conventional LB [120]

N	M	$\|\mathbf{x}^*\|_0$	NoI	$\frac{\|\mathbf{x}_p - \mathbf{x}^*\|}{\|\mathbf{x}^*\|}$	time (s)
4000	2000	$0.05N$	219	1.0299e-5	0.6
20000	10000		1008	9.7563e-6	9.1
50000	25000		759	9.5520e-6	19.0
4000	1000	$0.02N$	345	1.0664e-5	0.8
20000	5000		1727	9.2210e-6	14.7
50000	12500		1201	1.0275e-5	27.4

Table 4.2: Fast Dual-Based LB (Algorithm 4.2)

In addition, Fig. 4.1 illustrates the number of iterations for Algorithm 4.2 to achieve a precision of $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|/\|\mathbf{b}\| < 10^{-5}$ versus μ from 1 to 100 where the parameters were set to $N = 5 \times 10^4$, $M = 0.5N$, $K = 0.02N$. It is observed that the number of iterations increases approximately linearly with respect to μ. Unlike parameter λ involved in the ℓ_1-ℓ_2 unconstrained problem (2.12) that needs to be tuned diligently, Fig. 4.1 indicates that the number of iterations w.r.t. μ for a given solution accuracy is rather predictable. In effect, the iteration number required by Algorithm 4.2 for a highly accurate solution remains fairly small relative to that required by FISTA.

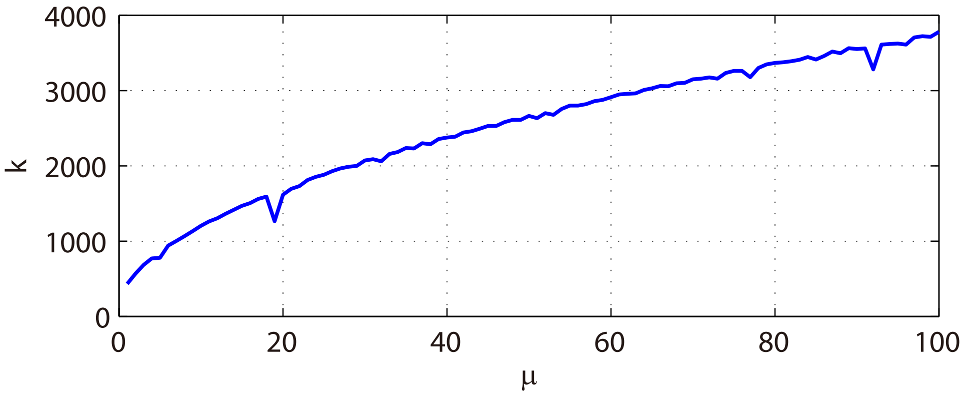


Figure 4.1: Number of iterations required by Algorithm 4.2 (with $N = 5 \times 10^4$, $M = 0.5N$, $K = 0.02N$) versus parameter μ.

4.4.2 Compressive Sensing of a Synthetic Image

To evaluate the proposed Algorithm 4.2 for large-scale data, we applied it to a test image $\mathbf{X}^*$ of size 512×512 (see Fig. 4.2(a)) which was produced by retaining its $K = 7 \times 10^3$ largest (9-level 2-D Haar) wavelet coefficients of an original image known as "`man`". Thus $\mathbf{X}^*$ is sparse in the wavelet domain as 97.33% of its wavelet coefficients are zero. Image $\mathbf{X}^*$ was then normalized so that its components are in between 0 and 1.

To apply Algorithm 4.2, we adopted a sampling matrix to measure the wavelet coefficients of the image. The measurement matrix $\mathbf{A}$ was a partial 2-D DCT matrix of size $M \times N$ with $M = \lceil 0.2N \rceil$ and $N = 512^2$. The M rows were chosen randomly from an $N \times N$ 2-D DCT matrix. We remark that $\mathbf{A}$ needs not to be explicitly produced or stored as any matrix-vector product involving $\mathbf{A}$ can be carried out by fast 2-D DCT. Parameter μ was set to 100. Note that $\|\mathbf{A}\mathbf{A}^T\| = 1$, hence $L = \mu$. The algorithm was terminated as soon as the relative constraint error $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|/\|\mathbf{b}\|$ falls below 10^{-2}. It took the proposed fast algorithm 217 iterations (41.1 seconds) to converge. The relative reconstruction error as measured by $(\|\mathbf{X}_p - \mathbf{X}^*\|_2)/\|\mathbf{X}^*\|_2$ was found to be 0.0116, where $\mathbf{X}_p$ represents the reconstructed image and $\|\cdot\|_2$ denotes the matrix Frobenius norm. By comparison, a total of 3168 iterations (601.6 seconds) were needed for the conventional LB algorithm to reconstruct an image with a relative reconstruction error 0.0118. The original and the reconstructed images are illustrated in Fig. 4.2. The visual difference between the two is hardly noticeable.

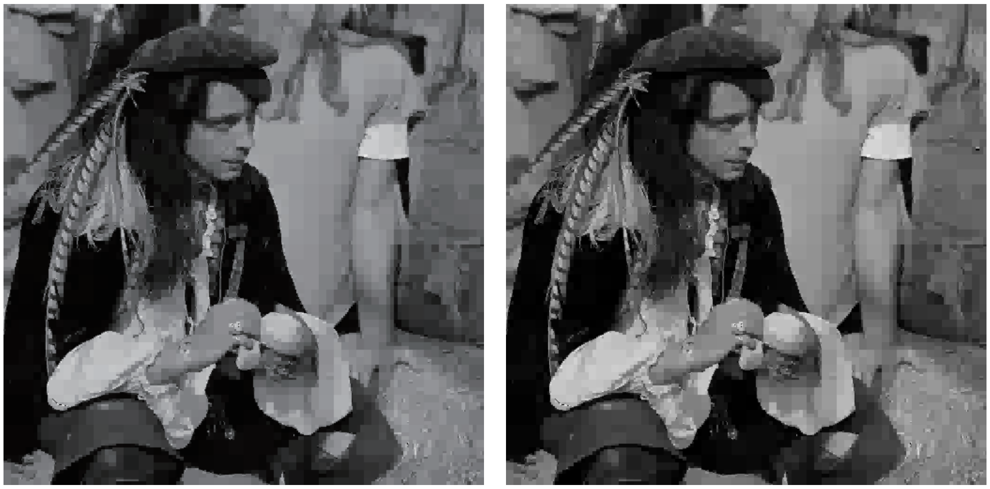


Figure 4.2: (a) Synthesized image "`man`" with 97.33% zero wavelet coefficients; (b) Reconstructed image "`man`" with 20% of DCT sampled coefficients by fast dual-based LB algorithm with 217 iterations.

4.4.3 Compressive Sensing of Natural Images

In the last set of experiments, we demonstrate efficiency of the proposed algorithm in solving large-scale problems by performing reconstructions of several 256 by 256 natural images based on compressive measurements. In the simulations, vector $\mathbf{x}^*$, the column version of the original image, is sampled by the following

$$\mathbf{b} = \mathbf{A}\mathbf{x}^*$$

The measurement matrix is defined as $\mathbf{A} = \mathbf{C}\boldsymbol{\Psi}^T$, with $\mathbf{C}$ the partial 2-D DCT matrix and $\boldsymbol{\Psi}$ corresponding to the (8-level 2-D Haar) inverse wavelet transform matrix. Since wavelet coefficients $\mathbf{s}$ of natural image $\mathbf{x}$ are sparse by $\mathbf{x} = \boldsymbol{\Psi}\mathbf{s}$, it follows that an ℓ_1 minimization problem can be formulated to reconstruct wavelet coefficient vector $\mathbf{s}$ in a CS framework

$$\begin{aligned} \text{minimize} \quad & ||\mathbf{s}||_1 && \text{(4.28a)} \\ \text{subject to:} \quad & \mathbf{A}\boldsymbol{\Psi}\mathbf{s} = \mathbf{b} && \text{(4.28b)} \end{aligned}$$

Furthermore, by substitution of $\mathbf{A} = \mathbf{C}\mathbf{\Psi}^T$, Problem (4.28) is equivalent to

$$\begin{aligned} \text{minimize} \quad & ||\mathbf{s}||_1 && (4.29a) \\ \text{subject to:} \quad & \mathbf{C}\mathbf{s} = \mathbf{b} && (4.29b) \end{aligned}$$

It can be observed that measurements $\mathbf{b}$ were essentially the same as the data obtained by sampling wavelet coefficients of the image under a partial 2-D DCT matrix.

In this experiment, partial 2-D DCT matrix $\mathbf{C}$ of size $M \times N$ was specified as $M = 20000$ and $N = 65536$. Parameter μ was set to 100. The proposed accelerated algorithm was applied to solve the large-scale optimization problem (4.29). In practice, matrix-vector multiplications involving big matrices like $\mathbf{C}$ or $\mathbf{C}^T$ are performed by fast 2-D DCT or inverse 2-D DCT. This makes it unnecessary to explicitly store $\mathbf{C}$ in memory. The algorithm was terminated when $\|\mathbf{C}\mathbf{s}^k - \mathbf{b}\|/\|\mathbf{b}\| < 10^{-2}$. After obtaining the global minimizer $\mathbf{s}_p$ of (4.29), an additional step was employed to reconstruct the image by

$$\mathbf{x}_p = \mathbf{\Psi}\mathbf{s}_p$$

The number of iterations (NoI), the relative reconstruction error, and the CPU time required for reconstruction of a number of digital images are listed in Tables 4.3 and 4.4 for the conventional LB method and the proposed fast algorithm, respectively. It can be seen that the proposed algorithm converges with number of iterations significantly less than those obtained from the conventional algorithm. As a matter of fact, it takes less than 10% of the time for the fast dual-based LB algorithm to achieve similar reconstruction performance compared with the conventional LB method. Figs. 4.3-4.8 illustrate reconstruction performance of 6 images by the proposed fast algorithm. Besides the SNRs computed, a visual inspection further supports that good reconstruction quality can be achieved with number of measurements less than 1/3 of size of image by the proposed fast algorithm rather efficiently in the CS framework.

In summary, we in this chapter have proposed a fast dual-based linearized Bregman algorithm. Our analysis is focused on the Lagrangian dual function for which a fast iterative scheme is developed in identifying the global minimizer of the problem. This method accelerates the linearized Bregman method and shares a convergence rate of $O(1/k^2)$. Experimental results are presented to demonstrate the superiority of the proposed algorithm compared with the conventional LB method for CS recovery of large-scale signals and images.

Images	**NoI**	$\frac{\|\mathbf{x}_p-\mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2}$	**time (s)**
cameraman	6951	0.0988	167.4
lena	7547	0.1311	185.1
barbara	6302	0.1721	147.6
fruits	6928	0.0716	163.9
boats	6641	0.1013	152.6
circles	3016	0.0112	69.1
building	5269	0.0695	117.0
crosses	7562	0.0146	173.8
bird	8654	0.0428	225.1

Table 4.3: Wavelet coefficients reconstruction by conventional LB

Images	**NoI**	$\frac{\|\mathbf{x}_p-\mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2}$	**time (s)**
cameraman	515	0.1014	14.7
lena	526	0.1335	12.7
barbara	478	0.1723	11.5
fruits	510	0.0751	12.1
boats	500	0.1039	12.2
circles	249	0.0097	6.0
building	438	0.0728	10.7
crosses	401	0.0126	9.8
bird	597	0.0475	14.7

Table 4.4: Wavelet coefficients reconstruction by fast dual-based LB

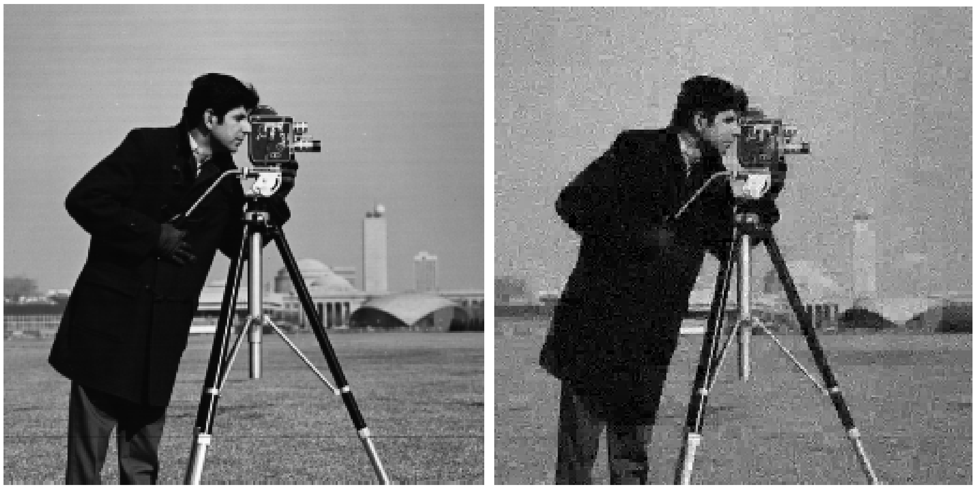


Figure 4.3: Left: image `cameraman`. Right: reconstructed image (SNR = 19.88 dB)

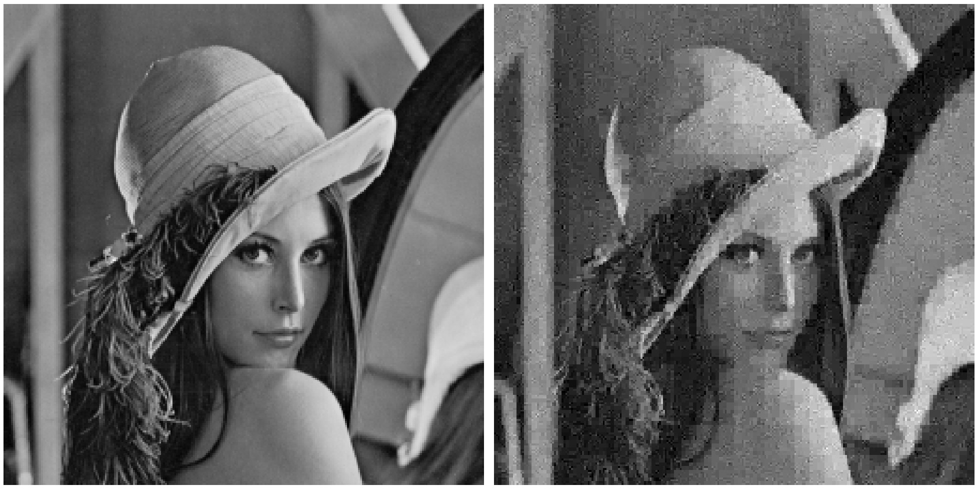


Figure 4.4: Left: image `lena`. Right: reconstructed image (SNR = 17.49 dB)

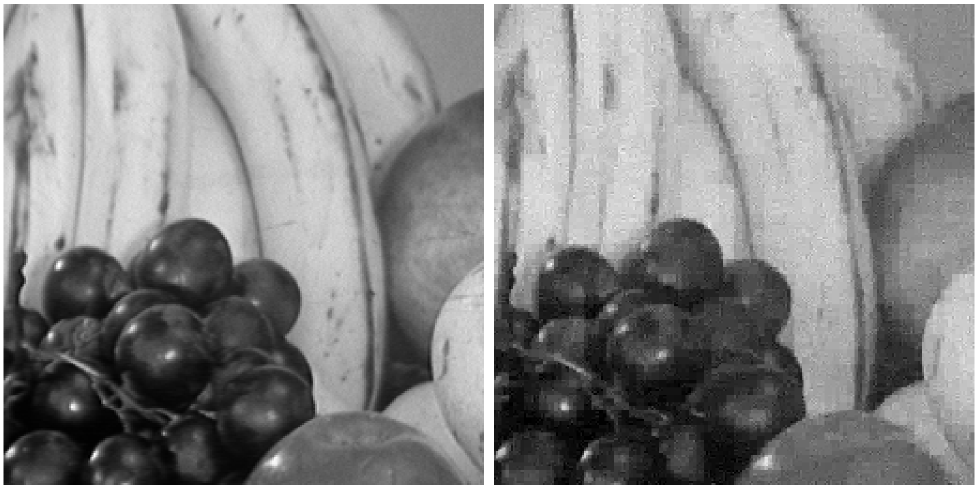


Figure 4.5: Left: image `fruits`. Right: reconstructed image (SNR = 22.48 dB)

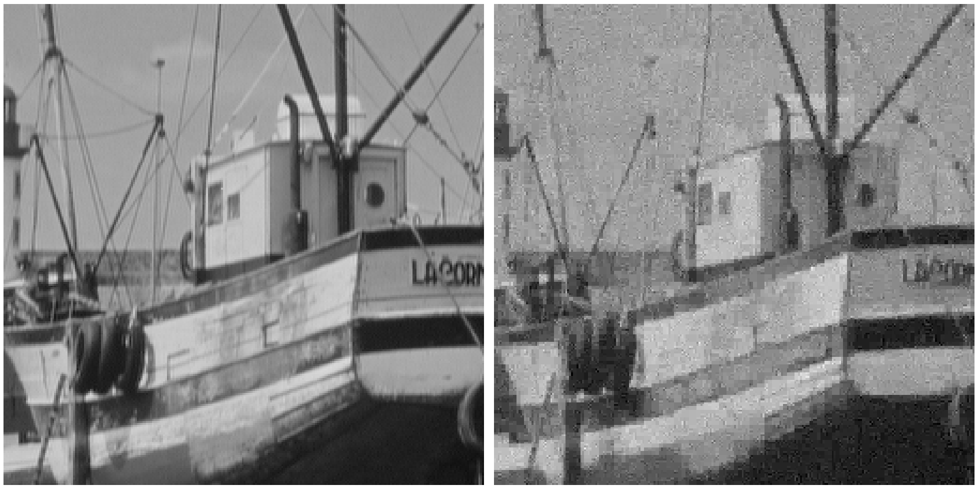


Figure 4.6: Left: image `boats`. Right: reconstructed image (SNR = 19.66 dB)

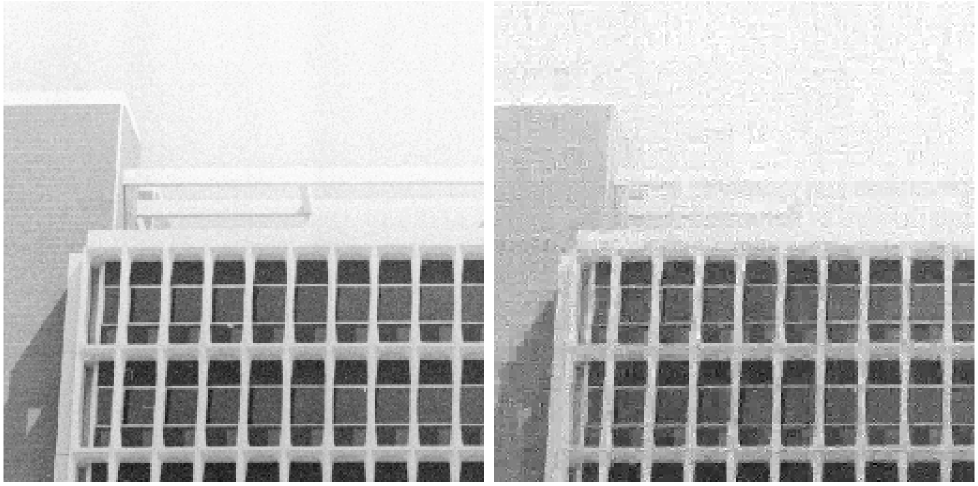


Figure 4.7: Left: image `building`. Right: reconstructed image (SNR = 22.76 dB)

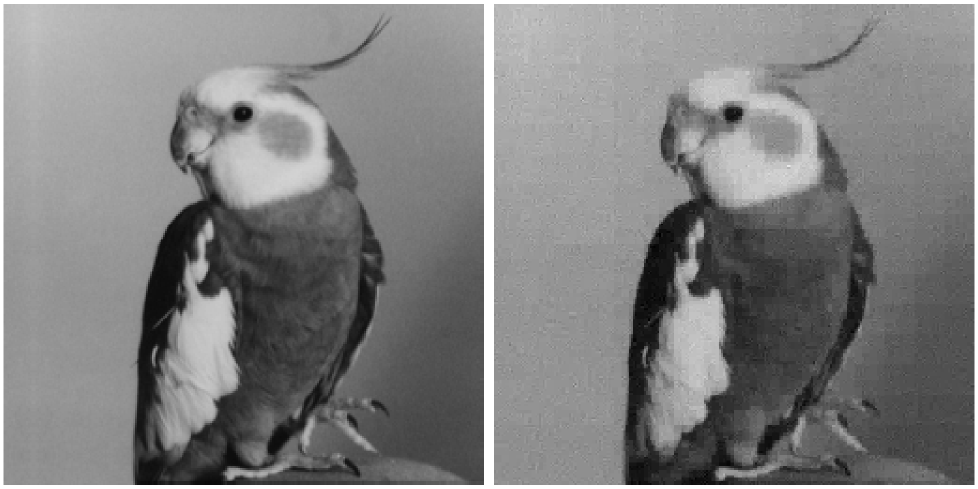


Figure 4.8: Left: image `bird`. Right: reconstructed image (SNR = 26.47 dB)

Chapter 5

Image Denoising by Generalized Total Variation Regularization

A breakthrough for denoising (especially piecewise smooth) images is made in the work of Rudin, Osher and Fatemi (ROF) [98] in which the standard ℓ_2-norm fidelity optimization is regularized by the total variation (TV) of the image. Variants of the ROF algorithm with improved performance and complexity are available [9, 28, 31]. In [32], a TV-ℓ_1 model for image denoising is proposed and the model is shown to be contrast invariant which in turn suggests a data driven scale selection technique, see also [2, 3] for in-depth mathematical analysis of TV-regularized denoising algorithms. Reference [111] deals with the denoising problem based on a model that involves a generalized TV$_q$ regularizer with $q \in [1, 2)$ and an ℓ_p-norm fidelity term with $p \geq 1$, and an iteratively-reweighted-norm algorithm is proposed to carry out the TV-ℓ_p minimization. Adaptive TV denoising has also been developed [38] in that the regularization term acts like a TV norm at object edges while approximating ℓ_2-norm in flat and ramp regions so as to avoid staircase effect. In [41], recovery of blocky images with improved performance over existing PDE-based approaches is addressed using a spatially adaptive TV model where a carefully designed penalization function is incorporated. Recently, the methodology for TV-based denoising is extended to a local TV-filtering scheme by employing the ROF model in a given neighbourhood of each pixel [75], and the concept of higher degree total variation (HDTV) is introduced in [68] in order to deal with the staircase and ringing artifacts that are common in TV and wavelet based schemes. It is also interesting to note that the anisotropic HDTV regularizer is found to provide consistently better reconstruction performance over the isotropic counterpart [68]. Very recently, a computational framework was proposed in [73] that incorporates a TV minimization formulation into a moving least squares method for image denoising in order to overcome drawbacks of both approaches.

In this chapter, we first generalize the standard TV to a pth-power TV with $0 \leq p \leq 1$ for further promoting gradient sparsity. Next a generalized TV (GTV) regularized least squares problem for image denoising is proposed. The GTV-regularized least squares problem is nonconvex, thus one contribution of this chapter is the development of a two-step solution method to solve the problem at hand - by introducing weighted TV (WTV) in which each element of discretized TV is weighted along the horizontal and vertical directions, then approximating the solution of the GTV-regularized problem by solving iteratively reweighted TV (IRTV) convex subproblems. In addition, new techniques are also developed to deal with technical difficulties encountered. These include a power-iterative warm-start technique for the proposed IRTV algorithm to reach a target power value $p_t < 1$ and a modified Split Bregman method in order to overcome the difficulties arising from the presence of the nontrivial weights in WTV. Numerical results are presented to demonstrate improved denoising performance in comparison with BPDN, IRL1, TV-ℓ_1 as well as several more recent denoising methods.

5.1 Generalized Total Variation Regularization

5.1.1 Generalized pth Power Total Variation

TV can be related to the ℓ_1 norm in two ways. Conceptually, both constrained ℓ_1-norm and TV minimization are found effective in signal processing, including denoising [39], deblurring [9, 80], and signal reconstruction [21]. The discretized anisotropic and isotropic TV of image $\mathbf{U}$ are given in Eqs. (2.17) and (2.18), respectively.

Analytically, the anisotropic total variation $\text{TV}^{(\text{A})}(\mathbf{U})$ may be interpreted as the ℓ_1 norm of the discretized gradient of $\mathbf{U}$ [21], namely, $\text{TV}^{(\text{A})}(\mathbf{U}) = \|\nabla_x \mathbf{U}\|_1 + \|\nabla_y \mathbf{U}\|_1$ where $(\nabla_x \mathbf{U})_{i,j} = U_{i,j} - U_{i+1,j}$, $(\nabla_y \mathbf{U})_{i,j} = U_{i,j} - U_{i,j+1}$ and $\|\cdot\|_1$ denotes the sum of magnitudes of the matrix's entries. Moreover, if we define $(D\mathbf{U})_{i,j} = (U_{i,j} - U_{i+1,j}) + \sqrt{-1}(U_{i,j} - U_{i,j+1})$, then the isotropic TV becomes $\|D\mathbf{U}\|_1$. In addition, when the size of $\mathbf{U}$ is reduced to $m \times 1$ or $1 \times n$, both $\text{TV}^{(\text{A})}(\mathbf{U})$ and $\text{TV}^{(\text{I})}(\mathbf{U})$ become $\|D\mathbf{U}\|_1$.

We now propose a generalized pth power total variation, denoted as TV_p, as follows. For the anisotropic case, $\mathrm{TV}_p^{(\mathrm{A})}$ is defined as

$$\mathrm{TV}_p^{(\mathrm{A})}(\mathbf{U}) = \sum_{i=1}^{m-1}\sum_{j=1}^{n} |U_{i,j} - U_{i+1,j}|^p + \sum_{i=1}^{m}\sum_{j=1}^{n-1} |U_{i,j} - U_{i,j+1}|^p \tag{5.1}$$

Note that $\mathrm{TV}_p^{(\mathrm{A})}(\mathbf{U})$ is related to ℓ_p "norm" by $\mathrm{TV}_p^{(\mathrm{A})}(\mathbf{U}) = \|\nabla_x \mathbf{U}\|_p^p + \|\nabla_y \mathbf{U}\|_p^p$. For the isotropic case, the generalized TV is defined by

$$\begin{aligned}\mathrm{TV}_p^{(\mathrm{I})}(\mathbf{U}) = &\sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \sqrt{|U_{i,j} - U_{i+1,j}|^{2p} + |U_{i,j} - U_{i,j+1}|^{2p}} \\ &+ \sum_{i=1}^{m-1} |U_{i,n} - U_{i+1,n}|^p + \sum_{j=1}^{n-1} |U_{m,j} - U_{m,j+1}|^p\end{aligned} \tag{5.2}$$

The value of power p in (2.17) and (2.18) is in the range $0 < p \leq 1$. Obviously, with $p = 1$, $\mathrm{TV}_p^{(\mathrm{A})}$ and $\mathrm{TV}_p^{(\mathrm{I})}$ recover the standard $\mathrm{TV}^{(\mathrm{A})}$ and $\mathrm{TV}^{(\mathrm{I})}$ respectively. We remark that reference [111] presents a TV_q-ℓ_p model for image restoration, hence is relevant to the TV_p model considered in this chapter. The main difference between the two is that the model in [111] is with $q \in [1, 2)$ and $p \geq 1$, hence it is a convex model, while the present paper investigates a nonconvex TV_p regularizer with $0 < p < 1$, which turns out to offer improved performance. We shall revisit this point in Sec. 5.3 where experimental results are reported.

With the generalized TV_p defined, we propose to study image denoising problem with a model that incorporates TV_p as the regularizer, namely,

$$\underset{\mathbf{U}}{\text{minimize}}\ \ \mathrm{TV}_p(\mathbf{U}) + \frac{\mu}{2}\|\mathbf{U} - \mathbf{B}\|_\mathrm{F}^2 \tag{5.3}$$

where $\mathrm{TV}_p(\mathbf{U})$ can be taken as $\mathrm{TV}_p^{(\mathrm{A})}(\mathbf{U})$ or $\mathrm{TV}_p^{(\mathrm{I})}(\mathbf{U})$ with $0 < p < 1$. With $p < 1$ solving the nonconvex problem in (5.3) is far from trivial and only local solutions can be retrieved. In what follows, an iterative reweighting technique is proposed to address the problem at hand.

5.1.2 Weighted TV and Iterative Reweighting

The idea of reweighting in the context of sparsity enhancement is originated in [24] where an iteratively reweighted ℓ_1-minimization (IRL1) technique is developed that connects an ℓ_0-regularized nonconvex problem directly to an ℓ_1-regularized convex problem. With $p < 1$ the generalized pth power TV proposed in Sec. 5.1.1 is also nonconvex. Here we propose an iterative reweighting which is tailored to tackle the TV_p-regularized denoising problem in (5.3) in a convex setting. As such, the proposed reweighting technique may be regarded as a natural extension of the method in [24] to models with total variation regularizers.

We proceed by introducing weighted TV (WTV), denoted by $\text{TV}_\text{w}(\mathbf{U})$, which for anisotropic TV is defined as

$$\text{TV}_\text{w}^{(\text{A})}(\mathbf{U}) = \sum_{i=1}^{m-1}\sum_{j=1}^{n} \alpha_{i,j}|U_{i,j} - U_{i+1,j}| + \sum_{i=1}^{m}\sum_{j=1}^{n-1} \beta_{i,j}|U_{i,j} - U_{i,j+1}| \tag{5.4}$$

and for isotropic TV is defined as

$$\begin{aligned}\text{TV}_\text{w}^{(\text{I})}(\mathbf{U}) =& \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \sqrt{(\alpha_{i,j}|U_{i,j} - U_{i+1,j}|)^2 + (\beta_{i,j}|U_{i,j} - U_{i,j+1}|)^2} \\ &+ \sum_{i=1}^{m-1}\alpha_{i,n}|U_{i,n} - U_{i+1,n}| + \sum_{j=1}^{n-1}\beta_{m,j}|U_{m,j} - U_{m,j+1}|\end{aligned} \tag{5.5}$$

In the above definitions, $\alpha_{i,j} > 0$ and $\beta_{i,j} > 0$ are weights to weigh the first order differences along the vertical and horizontal directions, respectively. Apparently $\text{TV}_\text{w}(\mathbf{U})$ becomes $\text{TV}(\mathbf{U})$ when all the weights are set to unity.

To see the critical role the TV_w plays in the new algorithm, notice that function $\text{TV}_\text{w}(\mathbf{U})$ remains convex as long as the weights are fixed. Now if in the lth round of $\text{TV}_\text{w}(\mathbf{U})$ minimization one assigns the weights to

$$\alpha_{i,j} = (|U_{i,j}^{(l)} - U_{i+1,j}^{(l)}| + \varepsilon)^{p-1}, \beta_{i,j} = (|U_{i,j}^{(l)} - U_{i,j+1}^{(l)}| + \varepsilon)^{p-1} \tag{5.6}$$

where $\varepsilon > 0$ is a small constant to prevent the weights from being zero, then for $\mathbf{U}$ in a small neighborhood of iterate $\mathbf{U}^{(l)}$ (5.4) and (5.6) in conjunction with the continuity of $\mathrm{TV_w}(\mathbf{U})$ imply that

$$\begin{aligned}
\mathrm{TV}_{\mathrm{w}}^{(\mathrm{A})}(\mathbf{U}) &\approx \mathrm{TV}_{\mathrm{w}}^{(\mathrm{A})}(\mathbf{U}^{(l)}) \\
&= \sum_{i=1}^{m-1}\sum_{j=1}^{n} \frac{|U_{i,j}^{(l)} - U_{i+1,j}^{(l)}|}{(|U_{i,j}^{(l)} - U_{i+1,j}^{(l)}| + \varepsilon)^{1-p}} + \sum_{i=1}^{m}\sum_{j=1}^{n-1} \frac{|U_{i,j}^{(l)} - U_{i,j+1}^{(l)}|}{(|U_{i,j}^{(l)} - U_{i,j+1}^{(l)}| + \varepsilon)^{1-p}} \\
&\approx \sum_{i=1}^{m-1}\sum_{j=1}^{n} |U_{i,j}^{(l)} - U_{i+1,j}^{(l)}|^p + \sum_{i=1}^{m}\sum_{j=1}^{n-1} |U_{i,j}^{(l)} - U_{i,j+1}^{(l)}|^p \\
&= \mathrm{TV}_{p}^{(\mathrm{A})}(\mathbf{U}^{(l)})
\end{aligned} \tag{5.7}$$

With a similar argument, we can also see that $\mathrm{TV}_{\mathrm{w}}^{(\mathrm{I})}(\mathbf{U}) \approx \mathrm{TV}_{p}^{(\mathrm{I})}(\mathbf{U}^{(l)})$ for $\mathbf{U}$ in a small vicinity of $\mathbf{U}^{(l)}$. Consequently, nonconvex minimization of $\mathrm{TV}_p(\mathbf{U})$ can practically be carried out by convex minimization of $\mathrm{TV_w}(\mathbf{U})$ with reweighting strategy (5.6) for each iteration. In other words, the introduction of weighted TV with the weights in (5.6) allows an effective convexification of the nonconvex TV_p. We note that in [75], the ROF model is examined in a given neighborhood of each image pixel with a local filter and weights are introduced to a local window to construct a locally weighted TV denoiser in order to reduce staircase effect, while the reweighting in the proposed algorithm is performed globally on the TV norm to enhance the image's gradient sparsity.

Based on above analysis, we propose an iteratively reweighted TV (IRTV) algorithm as Algorithm 5.1. The core of the proposed algorithm is the WTV-regularized problem (5.8) for given sets of weights $\{\alpha_{i,j}\}$ and $\{\beta_{i,j}\}$. We reiterate that problem (5.8) is convex. The algorithmic details of solving (5.8) for both weighted anisotropic and isotropic TV by Split Bregman type iterations are given in Sec. 5.2.

Algorithm 5.1 Algorithm for IRTV Regularized Minimization

```
1: Select parameters $\mu$, $p_t$ (a target value for power $p$) and $\varepsilon$. Set iteration count $l = 0$ and $\alpha_{i,j} = \beta_{i,j} = 1$.
2: Solve the WTV-regularized problem
       $$\underset{\mathbf{U}}{\text{minimize}} \ \mathrm{TV_w}(\mathbf{U}) + \frac{\mu}{2}\|\mathbf{U} - \mathbf{B}\|_{\mathrm{F}}^2 \tag{5.8}$$
   for $\mathbf{U}^{(l+1)}$.
3: With $\mathbf{U}^{(l+1)}$ update the weights $\{\alpha_{i,j}\}$ and $\{\beta_{i,j}\}$ in (5.6).
4: Terminate if $l + 1$ reaches a maximum number of iterations $L$. Otherwise, set $l = l + 1$ and repeat from step 2.
```

Implementation Issues

Suppose we are given a target value $p_t < 1$ for power p. Although the subproblem (5.8) involved in each round of iteration is convex, the TV$_p$-regularized problem with $p = p_t$ is nonconvex that admits many local minimizers.

The nonconvex nature of the problem at hand is reflected in Algorithm 5.1 by the way the initial weights $\{\alpha_{i,j}\}$ and $\{\beta_{i,j}\}$ are selected. The IRTV algorithm proposed in Sec. 5.1.2 is implemented using a power-iterative warm-start strategy, which may be explained by looking at the case of $p_t = 0.4$. The power-iterative strategy suggests that we start the algorithm with $p = 1$. So the problem at hand is convex and the solution is unique and global regardless of the initial point used. Now we use the solution just obtained as the initial point for a slightly reduced power value, say $p = 0.8$. Although TV$_p$-regularization with $p = 0.8$ is nonconvex, its global solution cannot be too far away from the solution with $p = 1$. So with an ℓ_1 solution as a "warm" initial point, the IRTV algorithm will converge to a reasonably good local solution if not the global solution. Once the second solution is obtained, it is used as the initial point for the problem in (5.8) with a further reduced power value, say $p = 0.6$. Doing the same thing one more time will reach the target power value $p = 0.4$. Numerical experiments have indicated that decreasing the power value each time by 0.2 is adequate to secure an excellent local solution for any target power between 0 and 1. Additional implementation details will be given in Sec. 5.3.

5.2 Split Bregman Type Iterations for the WTV-Regularized Problem

It turns out that efficient methods such as those based on Split Bregman (SB) iterations [60] are not applicable to (5.8) as long as nontrivial weights $\{\alpha_{i,j}\}$ and $\{\beta_{i,j}\}$ are present. This section describes algorithmic details of a technique for solving (5.8) for both the weighted anisotropic and isotropic TVs based on *modified* SB iterations where the standard SB iterations are revised in a major way to make the proposed algorithms work.

5.2.1 Solving (5.8) for Anisotropic TV

By the definition of anisotropic WTV in (5.4), the problem in (5.8) can be explicitly expressed as

$$\underset{\mathbf{U}}{\text{minimize}}\ \{\sum_{i=1}^{m-1}\sum_{j=1}^{n}\alpha_{i,j}|U_{i,j}-U_{i+1,j}|+\sum_{i=1}^{m}\sum_{j=1}^{n-1}\beta_{i,j}|U_{i,j}-U_{i,j+1}|+\frac{\mu}{2}\|\mathbf{U}-\mathbf{B}\|_{\mathrm{F}}^2\} \quad (5.9)$$

If we introduce $\mathbf{D}_x$ and $\mathbf{D}_y$ with each element in the vector defined as

$$(\mathbf{D}_x)_{i,j}=(\nabla_x^{\mathrm{w}}\mathbf{U})_{i,j}=\alpha_{i,j}(U_{i,j}-U_{i+1,j}) \quad (5.10a)$$

$$(\mathbf{D}_y)_{i,j}=(\nabla_y^{\mathrm{w}}\mathbf{U})_{i,j}=\beta_{i,j}(U_{i,j}-U_{i,j+1}) \quad (5.10b)$$

where $\nabla_x^{\mathrm{w}}\mathbf{U}$ and $\nabla_y^{\mathrm{w}}\mathbf{U}$ denote the weighted first-order differences along the x and y directions, respectively, then we can write

$$\|\mathbf{D}_x\|_1=\sum_{i=1}^{m-1}\sum_{j=1}^{n}\alpha_{i,j}|U_{i,j}-U_{i+1,j}|, \qquad \|\mathbf{D}_y\|_1=\sum_{i=1}^{m}\sum_{j=1}^{n-1}\beta_{i,j}|U_{i,j}-U_{i,j+1}| \quad (5.11)$$

Consequently, a splitting strategy can be applied using (5.10) and (5.11) to formulate the problem at hand as

$$\underset{\mathbf{U}}{\text{minimize}} \quad \{\|\mathbf{D}_x\|_1+\|\mathbf{D}_y\|_1+\frac{\mu}{2}\|\mathbf{U}-\mathbf{B}\|_{\mathrm{F}}^2\} \quad (5.12a)$$

$$\text{subject to:} \quad \mathbf{D}_x=\nabla_x^{\mathrm{w}}\mathbf{U},\ \mathbf{D}_y=\nabla_y^{\mathrm{w}}\mathbf{U} \quad (5.12b)$$

Enforcing the constraints in (5.12b), we obtain

$$\begin{aligned}\underset{\mathbf{U},\mathbf{D}_x,\mathbf{D}_y}{\text{minimize}}\ &\Big\{\|\mathbf{D}_x\|_1+\|\mathbf{D}_y\|_1+\frac{\mu}{2}\|\mathbf{U}-\mathbf{B}\|_{\mathrm{F}}^2\\ &+\frac{\lambda}{2}\|\mathbf{D}_x-\nabla_x^{\mathrm{w}}\mathbf{U}-\mathbf{E}_x^{(k)}\|_{\mathrm{F}}^2+\frac{\lambda}{2}\|\mathbf{D}_y-\nabla_y^{\mathrm{w}}\mathbf{U}-\mathbf{E}_y^{(k)}\|_{\mathrm{F}}^2\Big\}\end{aligned}$$

where $\mathbf{E}_x^{(k)}$ and $\mathbf{E}_y^{(k)}$ are updated through Bregman iterations as follows. In the kth iteration, we solve three subproblems to obtain $\mathbf{U}^{(k+1)}$, $\mathbf{D}_x^{(k+1)}$, and $\mathbf{D}_y^{(k+1)}$ as

$$\begin{aligned}\mathbf{U}^{(k+1)}=&\underset{\mathbf{U}}{\text{argmin}}\Big\{\frac{\mu}{2}\|\mathbf{U}-\mathbf{B}\|_{\mathrm{F}}^2\\ &+\frac{\lambda}{2}\|\nabla_x^{\mathrm{w}}\mathbf{U}+\mathbf{E}_x^{(k)}-\mathbf{D}_x^{(k)}\|_{\mathrm{F}}^2+\frac{\lambda}{2}\|\nabla_y^{\mathrm{w}}\mathbf{U}+\mathbf{E}_y^{(k)}-\mathbf{D}_y^{(k)}\|_{\mathrm{F}}^2\Big\}\end{aligned} \quad (5.13)$$

$$\mathbf{D}_x^{(k+1)} = \underset{\mathbf{D}_x}{\operatorname{argmin}} \|\mathbf{D}_x\|_1 + \frac{\lambda}{2}\|\mathbf{D}_x - \nabla_x^{\mathrm{w}}\mathbf{U}^{(k+1)} - \mathbf{E}_x^{(k)}\|_{\mathrm{F}}^2 \tag{5.14a}$$

$$\mathbf{D}_y^{(k+1)} = \underset{\mathbf{D}_y}{\operatorname{argmin}} \|\mathbf{D}_y\|_1 + \frac{\lambda}{2}\|\mathbf{D}_y - \nabla_y^{\mathrm{w}}\mathbf{U}^{(k+1)} - \mathbf{E}_y^{(k)}\|_{\mathrm{F}}^2 \tag{5.14b}$$

and $\mathbf{E}_x^{(k)}$ and $\mathbf{E}_y^{(k)}$ are updated by

$$\mathbf{E}_x^{(k+1)} = \mathbf{E}_x^{(k)} + \nabla_x^{\mathrm{w}}\mathbf{U}^{(k+1)} - \mathbf{D}_x^{(k+1)} \tag{5.15a}$$

$$\mathbf{E}_y^{(k+1)} = \mathbf{E}_y^{(k)} + \nabla_y^{\mathrm{w}}\mathbf{U}^{(k+1)} - \mathbf{D}_y^{(k+1)} \tag{5.15b}$$

Initially, $\mathbf{D}_x^{(0)} = \mathbf{D}_y^{(0)} = \mathbf{E}_x^{(0)} = \mathbf{E}_y^{(0)} = \mathbf{0}$.

The problems in (5.14) can be solved effectively by soft shrinkage [124] as

$$\mathbf{D}_x^{(k+1)} = \mathcal{T}_{1/\lambda}(\nabla_x^{\mathrm{w}}\mathbf{U}^{(k+1)} + \mathbf{E}_x^{(k)}) \tag{5.16a}$$

$$\mathbf{D}_y^{(k+1)} = \mathcal{T}_{1/\lambda}(\nabla_y^{\mathrm{w}}\mathbf{U}^{(k+1)} + \mathbf{E}_y^{(k)}) \tag{5.16b}$$

where $\mathcal{T}_{1/\lambda}$ applies pointwisely as

$$\mathcal{T}_{1/\lambda}(z) = \operatorname{sgn}(z) \cdot \max\{|z| - 1/\lambda, 0\} \tag{5.17}$$

Solving problem (5.13) is however far from trivial as the conventional method in [60] is not applicable to the WTV due to the presence of nontrivial weights $\{\alpha_{i,j}\}$ and $\{\beta_{i,j}\}$. The technique we propose to solve (5.13) starts by writing the first-order optimality condition for problem (5.13) as

$$\mu\mathbf{U} + \lambda\left[(\nabla_x^{\mathrm{w}})^T\nabla_x^{\mathrm{w}} + (\nabla_y^{\mathrm{w}})^T\nabla_y^{\mathrm{w}}\right]\mathbf{U} = \mathbf{C}^{(k)} \tag{5.18}$$

where

$$\mathbf{C}^{(k)} = \mu\mathbf{B} + \lambda[(\nabla_x^{\mathrm{w}})^T(\mathbf{D}_x^{(k)} - \mathbf{E}_x^{(k)}) + (\nabla_y^{\mathrm{w}})^T(\mathbf{D}_y^{(k)} - \mathbf{E}_y^{(k)})], \tag{5.19}$$

$(\nabla_x^{\mathrm{w}})^T$ and $(\nabla_y^{\mathrm{w}})^T$ denote the adjoint operators of ∇_x^{w} and ∇_y^{w}, respectively, and

$$\begin{aligned}&\left[(\nabla_x^{\mathrm{w}})^T\nabla_x^{\mathrm{w}}\mathbf{U}\right]_{i,j}\\ =&\begin{cases}\alpha_{1,j}^2(U_{1,j}-U_{2,j}) & \text{for } i=1, 1\le j\le n\\ -\alpha_{m-1,j}^2(U_{m-1,j}-U_{m,j}) & \text{for } i=m, 1\le j\le n\\ \alpha_{i,j}^2(U_{i,j}-U_{i+1,j})-\alpha_{i-1,j}^2(U_{i-1,j}-U_{i,j}) & \text{elsewhere}\end{cases}\end{aligned}$$

and

$$\begin{aligned}&\left[(\nabla_y^{\mathrm{w}})^T\nabla_y^{\mathrm{w}}\mathbf{U}\right]_{i,j}\\ =&\begin{cases}\beta_{i,1}^2(U_{i,1}-U_{i,2}) & \text{for } 1\le i\le m, j=1\\ -\beta_{i,n-1}^2(U_{i,n-1}-U_{i,n}) & \text{for } 1\le i\le m, j=n\\ \beta_{i,j}^2(U_{i,j}-U_{i,j+1})-\beta_{i,j-1}^2(U_{i,j-1}-U_{i,j}) & \text{elsewhere}\end{cases}\end{aligned}$$

As the matrix involved in the linear system in (5.18) is symmetric and diagonally dominant, using a standard argument (see Section 10.2 of [72]) the iterations generated by the Gauss-Seidel method as applied to (5.18) can be shown to converge to the solution of (5.18). From the above analysis, it follows that the linear system in (5.18) can be expressed componentwisely as

$$\begin{aligned}&[\mu+\lambda(\alpha_{i,j}^2+\alpha_{i-1,j}^2+\beta_{i,j}^2+\beta_{i,j-1}^2)]U_{i,j}=C_{i,j}^{(k)}+\\ &\lambda(\alpha_{i,j}^2U_{i+1,j}+\alpha_{i-1,j}^2U_{i-1,j}+\beta_{i,j}^2U_{i,j+1}+\beta_{i,j-1}^2U_{i,j-1})\end{aligned}$$

which naturally suggests a Gauss-Seidel iteration scheme for $U_{i,j}$ as

$$U_{i,j}^{(r+1)}=\frac{\rho_{i,j}^{(r,r+1)}}{\tau_{i,j}}\qquad \text{for } r=0,1,\cdots,R \tag{5.20}$$

where

$$\begin{aligned}&\rho_{i,j}^{(r,r+1)}=C_{i,j}^{(k)}+\\ &\lambda\left[\alpha_{i,j}^2U_{i+1,j}^{(r)}+\alpha_{i-1,j}^2U_{i-1,j}^{(r+1)}+\beta_{i,j}^2U_{i,j+1}^{(r)}+\beta_{i,j-1}^2U_{i,j-1}^{(r+1)}\right]\end{aligned}$$

$$\tau_{i,j}=\mu+\lambda(\alpha_{i,j}^2+\alpha_{i-1,j}^2+\beta_{i,j}^2+\beta_{i,j-1}^2)$$

In words, $U_{i,j}^{(r+1)}$ is computed using iterates $U_{i+1,j}^{(r)}$, $U_{i,j+1}^{(r)}$ from the preceding iteration while iterates $U_{i-1,j}^{(r+1)}$, $u_{i,j-1}^{(r+1)}$ are computed from the current iteration. The convergence of the proposed Gauss-Seidel iterations in (5.20) is guaranteed by the diagonal dominance of the linear system involved.

5.2.2 Solving (5.8) for Isotropic TV

We now consider solving the convex problem in (5.8) with weighted isotropic TV (see (5.5) for its definition) using the Split Bregman technique. The weighted TV in this case can be written as $\mathrm{TV}_{\mathrm{w}}^{(\mathrm{I})}(\mathbf{U}) = \|(\mathbf{D}_x, \mathbf{D}_y)\|_F$ where

$$\|(\mathbf{D}_x, \mathbf{D}_y)\|_F = \sum_{i,j} \sqrt{(\mathbf{D}_x)_{i,j}^2 + (\mathbf{D}_y)_{i,j}^2}$$

By using the splitting strategy, the problem at hand can be formulated as

$$\begin{aligned} &\underset{\mathbf{U}}{\text{minimize}} && \|(\mathbf{D}_x, \mathbf{D}_y)\|_F + \frac{\mu}{2}\|\mathbf{U} - \mathbf{B}\|_{\mathrm{F}}^2 && (5.21\text{a}) \\ &\text{subject to:} && \mathbf{D}_x = \nabla_x^{\mathrm{w}}\mathbf{U},\ \mathbf{D}_y = \nabla_y^{\mathrm{w}}\mathbf{U} && (5.21\text{b}) \end{aligned}$$

By enforcing the equality constraint in (5.21b), the problem becomes

$$\begin{aligned} \underset{\mathbf{U},\mathbf{D}_x,\mathbf{D}_y}{\text{minimize}} \Big\{ &\|(\mathbf{D}_x, \mathbf{D}_y)\|_F + \frac{\mu}{2}\|\mathbf{U} - \mathbf{B}\|_{\mathrm{F}}^2 \\ &+\frac{\lambda}{2}\|\mathbf{D}_x - \nabla_x^{\mathrm{w}}\mathbf{U} - \mathbf{e}_x^{(k)}\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|\mathbf{D}_y - \nabla_y^{\mathrm{w}}\mathbf{U} - \mathbf{e}_y^{(k)}\|_{\mathrm{F}}^2 \Big\} \end{aligned} \tag{5.22}$$

Note that unlike the weighted anisotropic TV, here the variables $\mathbf{D}_x$ and $\mathbf{D}_y$ are coupled. We propose to tackle the problem as follows. In the kth iteration, we find a set of new iterate $\{\mathbf{U}^{(k+1)}, \mathbf{D}_x^{(k+1)}, \mathbf{D}_y^{(k+1)}\}$ for the problem in (5.22) in two steps: First, with fixed $\mathbf{D}_x^{(k)}$ and $\mathbf{D}_y^{(k)}$ we solve the subproblem

$$\begin{aligned} \mathbf{U}^{(k+1)} = &\underset{\mathbf{U}}{\text{argmin}} \Big\{ \frac{\mu}{2}\|\mathbf{U} - \mathbf{B}\|_{\mathrm{F}}^2 \\ &+\frac{\lambda}{2}\|\nabla_x^{\mathrm{w}}\mathbf{U} + \mathbf{E}_x^{(k)} - \mathbf{D}_x^{(k)}\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|\nabla_y^{\mathrm{w}}\mathbf{U} + \mathbf{E}_y^{(k)} - \mathbf{D}_y^{(k)}\|_{\mathrm{F}}^2 \Big\} \end{aligned} \tag{5.23}$$

Then we fix $\mathbf{U} = \mathbf{U}^{(k+1)}$ in (5.22) and solve it for $\mathbf{D}_x^{(k+1)}$ and $\mathbf{D}_y^{(k+1)}$, namely,

$$\begin{aligned}(\mathbf{D}_x^{(k+1)}, \mathbf{D}_y^{(k+1)}) = \underset{\mathbf{D}_x,\mathbf{D}_y}{\operatorname{argmin}} \Big\{ &\|(\mathbf{D}_x, \mathbf{D}_y)\|_F + \frac{\lambda}{2}\|\mathbf{D}_x - \nabla_x^{\mathrm{w}}\mathbf{U}^{(k+1)} - \mathbf{E}_x^{(k)}\|_{\mathrm{F}}^2 \\ &+ \frac{\lambda}{2}\|\mathbf{D}_y - \nabla_y^{\mathrm{w}}\mathbf{U}^{(k+1)} - \mathbf{E}_y^{(k)}\|_{\mathrm{F}}^2 \Big\}\end{aligned} \tag{5.24}$$

Despite the fact that the variables $\mathbf{D}_x$ and $\mathbf{D}_y$ are coupled, the subproblem (5.24) above can still be explicitly solved using a generalized shrinkage formula [60, 110]

$$\mathbf{D}_x^{(k+1)} = \max\,(\mathbf{S}^{(k)} - 1/\lambda, 0)\frac{\nabla_x^{\mathrm{w}}\mathbf{U}^{(k+1)} + \mathbf{E}_x^{(k)}}{\mathbf{S}^{(k)}} \tag{5.25a}$$

$$\mathbf{D}_y^{(k+1)} = \max\,(\mathbf{S}^{(k)} - 1/\lambda, 0)\frac{\nabla_y^{\mathrm{w}}\mathbf{U}^{(k+1)} + \mathbf{E}_y^{(k)}}{\mathbf{S}^{(k)}} \tag{5.25b}$$

where the divisions are pointwisely operated and

$$\mathbf{S}^{(k)} = \sqrt{|\nabla_x^{\mathrm{w}}\mathbf{U}^{(k+1)} + \mathbf{E}_x^{(k)}|^2 + |\nabla_y^{\mathrm{w}}\mathbf{U}^{(k+1)} + \mathbf{E}_y^{(k)}|^2 + \varepsilon^2}$$

where the term ε^2 is to prevent the elements of $\mathbf{S}^{(k)}$ from being zeros. On comparing the modified SB iteration for weighted isotropic TV with its anisotropic counterpart, the main difference is the way the next iterate $\{\mathbf{D}_x^{(k+1)}, \mathbf{D}_y^{(k+1)}\}$ is obtained: the anisotropic case uses standard soft shrinkage while the isotropic case requires general shrinkage formulas.

5.3 Experimental Studies

In this section, we apply the IRTV algorithm proposed in Secs. 5.1 and 5.2 to a variety of synthetic and natural images and its performance was evaluated in comparison with several algorithms in the literature that are known to be effective for image denoising. As argued in Sec. 5.1.2, both $\mathrm{TV}^{(\mathrm{A})}$ and $\mathrm{TV}^{(\mathrm{I})}$ are closely related to the ℓ_1 norm of the image gradient. As such, denoising performance based on them are expected to be similar to each other. As a matter of fact, in [9], both TVs were found to yield very similar results, while the simulations reported in [68] were slightly in favor of $\mathrm{TV}^{(\mathrm{A})}$. Under these circumstances, the simulations reported below were carried out using anisotropic type of TV and TV_p.

The denoising performance was evaluated by *peak signal-to-noise* ratio (PSNR) which is defined by

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right)\ (\mathrm{dB}) \tag{5.26}$$

with

$$\mathrm{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[U_{i,j} - U^*_{i,j}]^2 \tag{5.27}$$

where $\{U_{i,j}\}$ and $\{U^*_{i,j}\}$ denote the processed and desired (clean) images, respectively. Note that MSE (5.27) represents the variance of the noise remained in the processed image. Consequently, if the variance of the residual noise can be estimated (e.g. using a certain patch of the denoised image), PSNR remains to be a feasible measure in case the clean image is not available.

In what follows, the term "standard TV" (as well as term "TV" in Table 5.1) is referred to as an algorithm that solves problem (2.20). In our simulations, this problem was solved by the Split Bregman method [60]. Throughout the chapter, the term "TV-ℓ_1" is referred to as an algorithm that solves the problem

$$\underset{\mathbf{U}}{\text{minimize}}\ \ \mathrm{TV}(\mathbf{U}) + \mu\|\mathbf{U} - \mathbf{b}\|_1 \tag{5.28}$$

Since (5.28) is convex, in our simulations (5.28) was solved using CVX [63, 64].

5.3.1 Denoising with Different Rounds of Reweighting

In this part of experimental studies, effectiveness of the reweighting strategy (5.6) on denoising performance was examined. Shepp-Logan phantom of size 256 × 256 was chosen as the test image in the study, where the image was normalized to within $[0, 1]$ and corrupted with zero-mean additive white Gaussian noise with standard deviation 0.1. The PSNR of the noisy phantom was found to be 20.00 dB. To test the reweighting performance, power p was fixed to 0.9, ϵ was set to 10^{-3}, and μ in (5.8) varied in the range $[4, 25]$. For each μ in that range, λ was set to 2μ and the maximum number of Bregman iterations was set to 30.

For the first round of iterations of the IRTV algorithm, i.e., $l = 0$, the weights were all set to one, and the image computed from (5.8) simply corresponds to the standard TV denoising result. The weights in the following rounds of iterations were updated using (5.6), and the WTV-regularized problem was solved accordingly. Note that in solving (5.8) at the $(l+1)$th iteration, we passed the previous iterate $\mathbf{U}^{(l)}$ as the initial point $\mathbf{U}_0$. The IRTV algorithm was applied for each μ with the number of reweighting rounds set to 1, 2, and 3, respectively. The three PSNR-versus-μ curves together with the PSNR curve of the standard TV denoising (i.e., $l = 0$) are depicted in Fig. 5.1. We observe that the algorithm practically converges in 3 rounds of reweighting, yielding a peak PSNR of 36.44 dB relative to 34.32 dB offered by the conventional TV denoising. Also note that the first round of reweighting (i.e., $l = 1$) yields largest performance gain.

5.3.2 Denoising with Different Power p

Next, the IRTV algorithm was applied to the same phantom image with the value of power p reduced from 1 by 0.2 each time down to 0. The experiment's set-up was the same as in Experiment 1 except that the range of μ was extended to $[4, 50]$ to cope with the wide range of power values, and only one round of reweighting was used for each given power p since it provides most performance gain (see Sec. 5.3.1). In the experiment, the IRTV algorithm for a given power value p starts with the solution obtained with a slightly larger p as an initial point so as to facilitate the algorithm to converge to a satisfactory solution with a small number of iterations.

The PSNR curves associated with $p = 1, 0.8, 0.6, 0.4, 0.2$ and 0 are plotted in Fig. 5.2. We observe that the algorithm yields better performance for smaller power p, with the best performance of 39.39 dB achieved when $p = 0$ and μ set to 34 that is considerably higher than the peak PSNR of 34.32 dB by the standard TV denoising. The denoised images by the standard TV and IRTV algorithms are depicted in Fig. 5.3(a) and (c), respectively. To visualize the performance difference between the two algorithms, the difference between the original and denoised phantoms from the standard TV and IRTV algorithms are shown in Fig. 5.3(b) and (d), respectively. It is noticed that the difference after IRTV denoising is considerably less visible relative to that by the standard TV-based denoising.

5.3.3 Denoising of Natural and Synthetic Images

The proposed algorithm was tested on a variety of images of size 256 by 256 for different power p. Each image was normalized to be within $[0, 1]$, and was corrupted by zero-mean Gaussian noise with standard deviation 0.1, which corresponds to a PSNR of 20.00 dB.

For comparison purposes, several known denoising methods were also applied to the same set of test images. These include the well-known basis-pursuit denoising

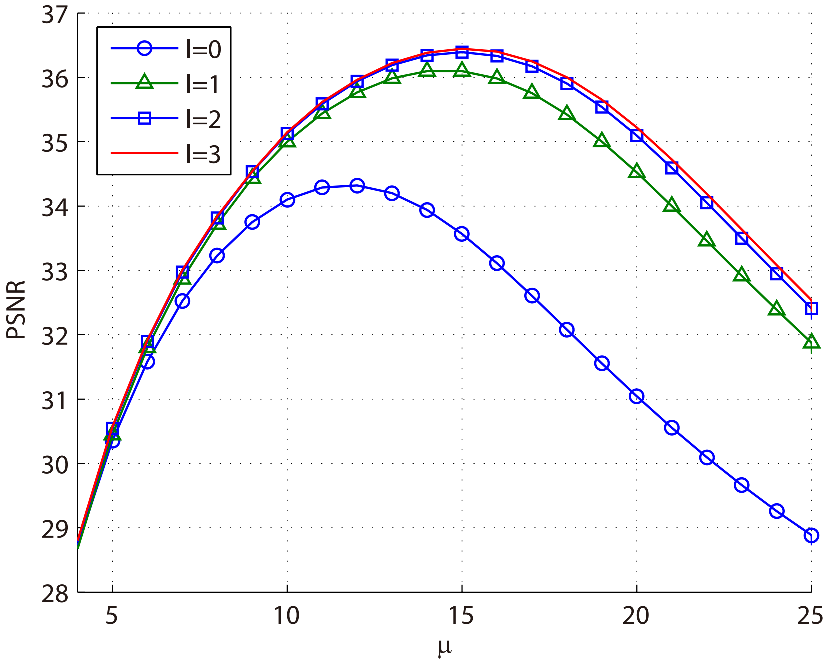


Figure 5.1: Denoising Shepp-Logan phantom with 3 rounds of reweighting for $p = 0.9$, compared with the standard TV denoising which corresponds to the curve with $l = 0$.

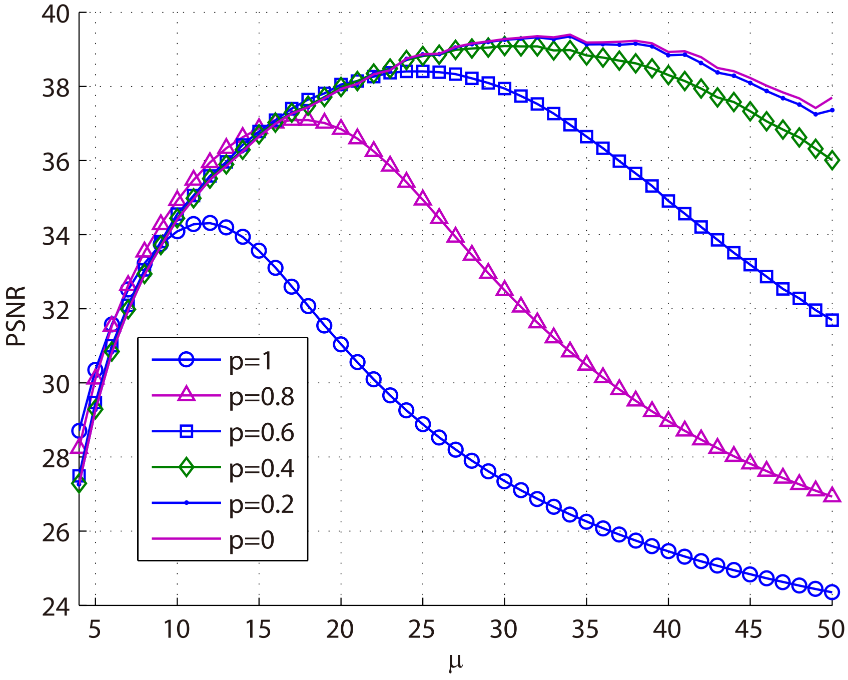


Figure 5.2: Denoising Shepp-Logan phantom with one round of reweighting and $p = 1, 0.8, 0.6, 0.4, 0.2$ and 0. The PSNR with $p = 1$ coincides the standard TV denoising.

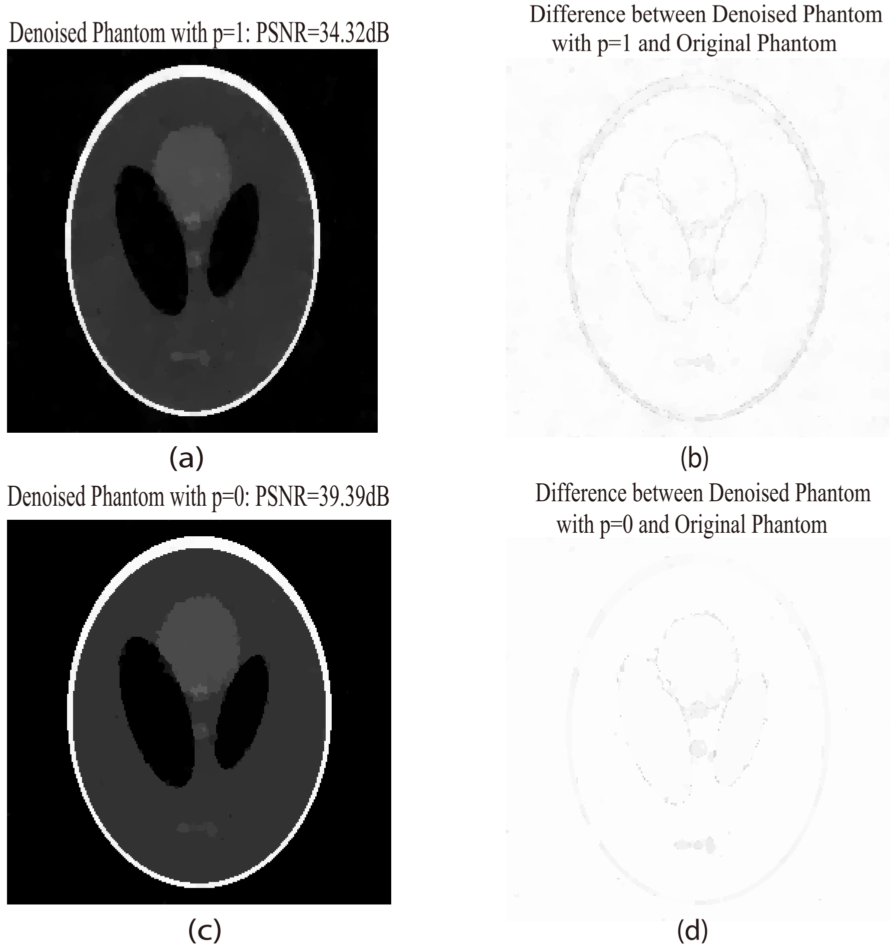

Denoised Phantom with p=1: PSNR=34.32dB
Difference between Denoised Phantom with p=1 and Original Phantom
(a)
(b)
Denoised Phantom with p=0: PSNR=39.39dB
Difference between Denoised Phantom with p=0 and Original Phantom
(c)
(d)

Figure 5.3: Denoising Shepp-Logan phantom (a) by the standard TV minimization (i.e. $p = 1$), and (c) by the IRTV algorithm with $p = 0$. Difference between the denoised and original images are shown in (b) for the standard TV and (d) for IRTV.

(BPDN) [39], iteratively reweighted ℓ_1 algorithm (IRL1) [24], ℓ_2-ℓ_p minimization based denoising [35,36,57,114,116], the TV-ℓ_1 model [32], and the standard TV-regularized algorithm. For fair comparisons, when an algorithm was applied, the parameter(s) involved were tuned to optimize the performance of that particular algorithm. As was in Sec. 5.3.1 and 5.3.2, here the proposed IRTV algorithm was implemented using the power-iterative and warm-start strategy described in Secs. 5.1.2. The performance of the five algorithms in terms of PSNR before and after denoising for a total of sixteen natural and synthetic test images are shown in Table 5.1. For each test image, the best PSNR achieved is indicated with a bold-faced number. The pair of numbers in the columns ℓ_p-ℓ_2/p and IRTV/p denote the PSNR (in dB) achieved and the value of power p utilized to obtain that PSNR, respectively. For visual inspection, Figs.

5.4-5.8 depict the IRTV-denoised natural images of "`axe`", "`church`", "`jet`", as well as a synthetic image "`phantomdisk`" and texture image "`fence`", respectively, as compared with their original versions and noisy counterparts.

Images	PSNR in dB					
	IRL1	BPDN	ℓ_p-ℓ_2/p	TV-ℓ_1	TV	IRTV/p
`axe`	25.22	28.51	28.77/0.6	31.09	33.03	**33.21**/0.6
`building`	21.57	25.42	25.54/0.0	26.47	27.85	**28.08**/0.8
`bird`	24.92	28.07	28.20/0.4	30.33	31.87	**31.94**/0.6
`circles`	24.05	28.35	28.71/0.2	33.87	37.35	**43.65**/0.2
`church`	21.43	25.34	25.49/0.6	27.10	28.42	**28.60**/0.6
`camera`	21.83	25.46	25.54/0.4	26.02	27.74	**27.98**/0.8
`crosses`	21.45	26.33	26.39/0.8	23.54	29.23	**32.20**/0.4
`dome`	20.00	22.47	19.68/0.6	18.18	24.00	**24.22**/0.8
`fence`	21.55	25.40	25.52/0.2	26.75	28.24	**28.45**/0.8
`jet`	21.40	24.96	25.03/0.4	26.05	27.45	**27.62**/0.8
`liftingbody`	25.63	28.56	28.71/0.4	30.49	32.06	**32.27**/0.8
`phantomdisk`	23.38	27.23	27.51/0.2	31.12	32.87	**33.20**/0.8
`pool`	22.84	25.83	25.93/0.6	27.45	28.55	**28.62**/0.6
`reschart`	20.56	25.56	25.86/0.4	25.45	29.11	**30.51**/0.8
`satellite`	21.69	25.50	25.63/0.4	25.95	27.95	**28.36**/0.8
`squares`	40.62	40.58	46.36/0.0	37.20	42.44	**47.41**/0.0
`tower`	22.93	25.78	25.86/0.4	26.74	27.88	**27.91**/0.6
`text`	20.36	25.88	26.22/0.6	27.31	30.38	**33.84**/0.0

Table 5.1: PSNRs of test images denoised by the proposed algorithm and several existing denoising algorithms. In each row, the boldfaced numerical value indicates the best PSNR for the image.

Based on the experimental results, we remark that

(1) The IRTV algorithm consistently outperforms the other algorithms. Especially for the synthetic images such as "`circles`", "`squares`" and "`text`", its performance gain over the second best performer, the standard TV denoising, is found to be significant. We note that for these images the best power p is zero or close to zero, while the standard TV algorithm corresponds to the case of $p = 1$. For natural images, although IRTV is found to offer decent gains over the standard TV for some images such as "`satellite`" (0.41 dB) and "`camera`" (0.24 dB), the improvement over natural images are obviously less impressive relative to that achieved for synthetic images. The reason for this is that for most natural images the optimal power p, is found to

be fairly close to 1, which implies that for natural images the standard TV denoising performs nearly optimally. In this regard the proposed IRTV algorithm tries to push the envelop to offer even better performance.

(2) The IRL1 algorithm does not seem to perform well relative to the other algorithms evaluated. There are several reasons for the outcome. Recall that the IRL1 essentially performs ℓ_0-regularization without leaving convex programming environment. As argued earlier, however, this is not suited for denoising natural images as the best power for these image appears to be close to 1. This explains why the BPDN, which is an ℓ_1-regularized algorithm, outperforms IRL1 for most instances. For the synthetic images, the IRL1 was expected to perform well. The reason it fails to do so has to do with its way to perform ℓ_0-regularization - it jumps from ℓ_1-regularization directly to ℓ_0-regularization as opposed to the proposed IRTV algorithm where TV$_p$ regularization is implemented with a power-iterative warm-start strategy. As a result, it leads to a suboptimal solution with degraded performance.

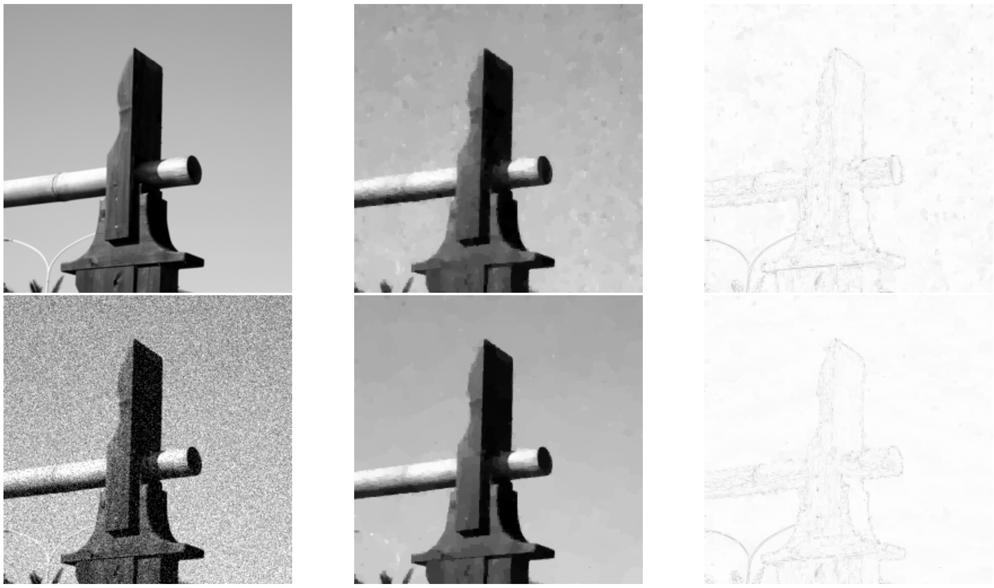


Figure 5.4: Denoising `axe` (up to bottom, left to right) (a) original image (b) noisy image (c) denoised image by the TV-ℓ_1 (d) denoised image by the IRTV (e) difference image between the TV-ℓ_1 denoised image and original image (f) difference image between the IRTV denoised image and original image

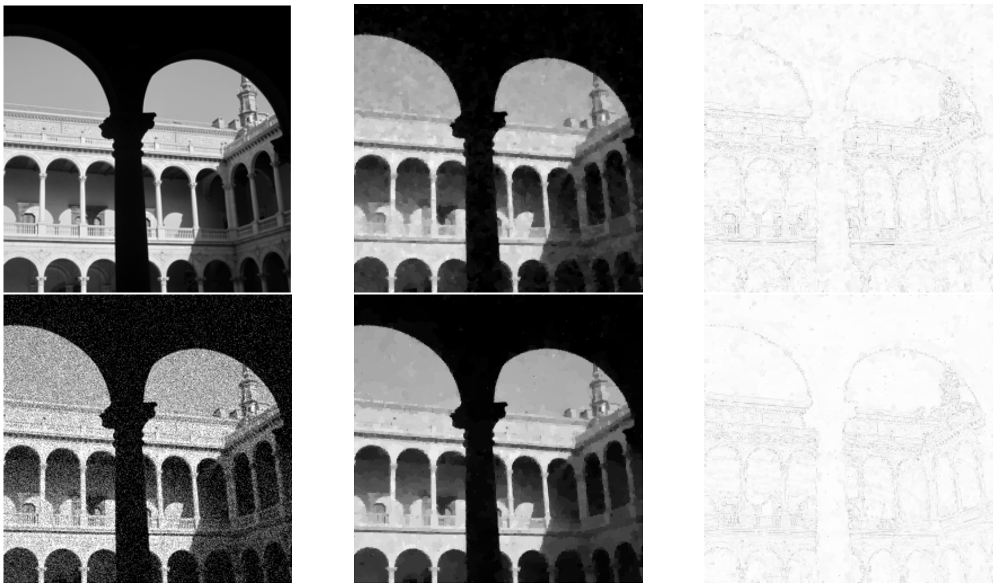


Figure 5.5: Denoising `church` (up to bottom, left to right) (a) original image (b) noisy image (c) denoised image by the TV-ℓ_1 (d) denoised image by the IRTV (e) difference image between the TV-ℓ_1 denoised image and original image (f) difference image between the IRTV denoised image and original image

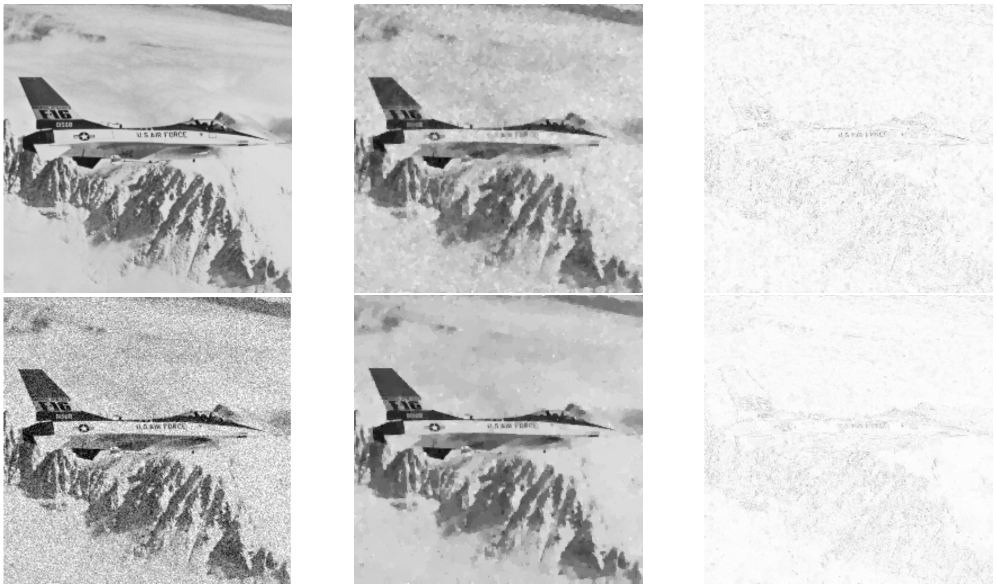


Figure 5.6: Denoising `jet` (up to bottom, left to right) (a) original image (b) noisy image (c) denoised image by the TV-ℓ_1 (d) denoised image by the IRTV (e) difference image between the TV-ℓ_1 denoised image and original image (f) difference image between the IRTV denoised image and original image

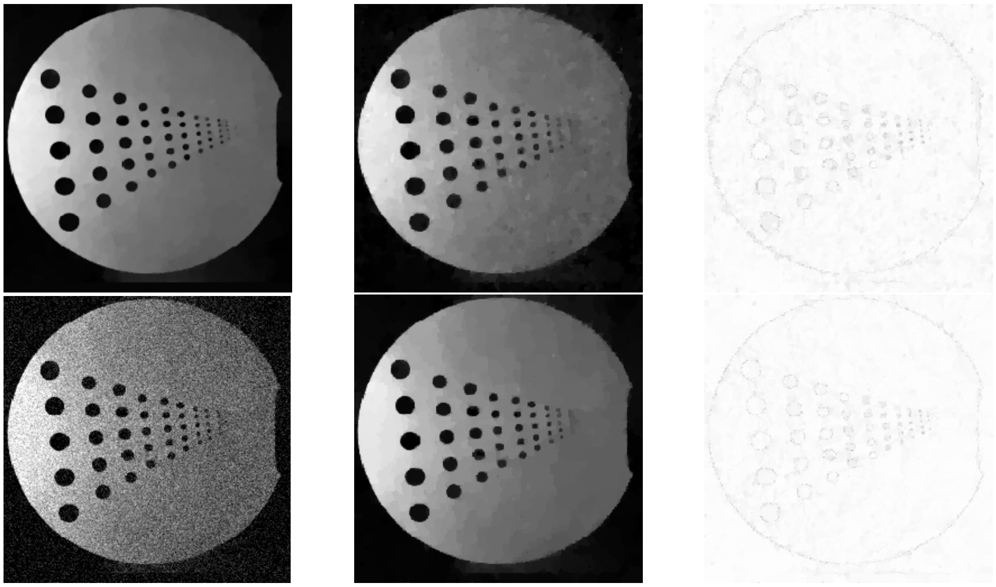


Figure 5.7: Denoising `phantomdisk` (up to bottom, left to right) (a) original image (b) noisy image (c) denoised image by the TV-ℓ_1 (d) denoised image by the IRTV (e) difference image between the TV-ℓ_1 denoised image and original image (f) difference image between the IRTV denoised image and original image

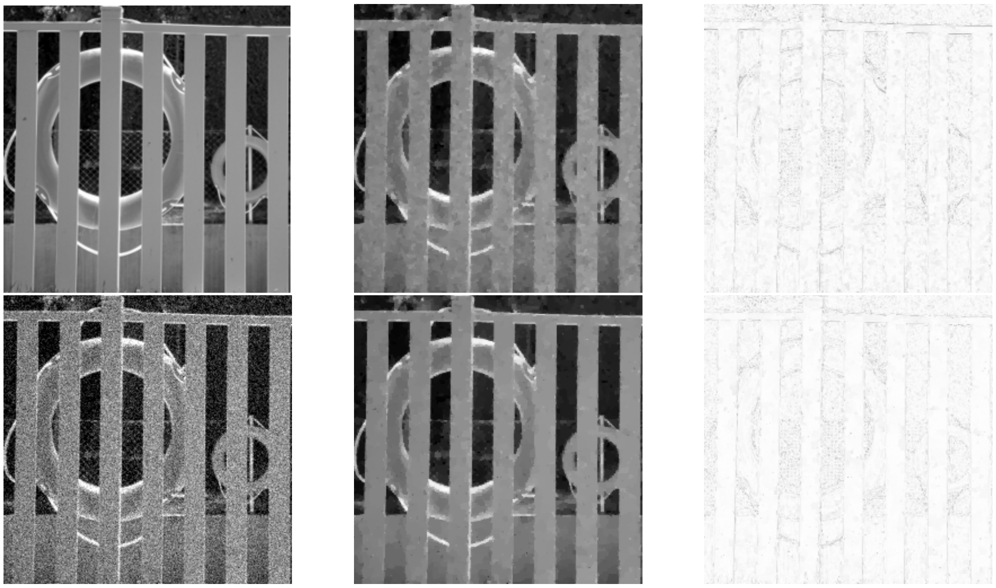


Figure 5.8: Denoising `fence` (up to bottom, left to right) (a) original image (b) noisy image (c) denoised image by the TV-ℓ_1 (d) denoised image by the IRTV (e) difference image between the TV-ℓ_1 denoised image and original image (f) difference image between the IRTV denoised image and original image

(3) By visual inspection of Figs. 5.4-5.8, the IRTV algorithm is found to well preserve edges as well as textures (see Fig. 5.8) relative to the well-established TV-ℓ_1 algorithm [32] that uses ℓ_1-norm as a measure of fidelity. It was argued that the TV-ℓ_1 model offers several advantages: the ℓ_1 regularization is more geometric, the TV-ℓ_1 model is contrast invariant, and the ℓ_1-regularized model suggests a data-driven scale selection mechanism. Experimental results on TV-ℓ_1 denoising and deblurring were reported in [111]. Note that in this regard IRTV also performs better than the TV-ℓ_1 algorithm as can be seen from Figs. 5.4-5.8.

Chapter 6

Compressive Imaging by Generalized Total Variation Regularization

Compressive imaging (CI) is a natural branch of compressive sensing (CS). The design of efficient CI system remains a challenging problem as it involves a large amount of data, which has far-reaching implications for the complexity of the optical design, calibration, data storage and computational burden. A step towards overcoming the memory requirements is accomplished recently in [52, 95] with development of CI methods by employing a separable sensing matrix. In [95], a two-dimensional separable sensing operator is used to reduce the storage of matrices of size $m^2 \times n^2$ produced by the Kronecker product of two $m \times n$ matrices. Another example is the work in [102] on TV minimization using a Split Bregman approach where a separable sensing operator is utilized.

The exact formulation of the CI optimization problem depends on the application being considered. For demonstration purposes, we focus on the application of CI to sparse magnetic resonance imaging (MRI). The general form for the sparse MRI reconstruction problem is presented and discussed in [34, 78]. We recall the formulation in Eq. (2.21) given by

$$\begin{array}{rl} \underset{\mathbf{U}}{\text{minimize}} & \text{TV}(\mathbf{U}) \\ \text{subject to:} & \|\mathbf{R} \circ (\mathcal{F}\mathbf{U}) - \mathbf{B}\|_{\text{F}}^2 < \sigma^2 \end{array}$$

where $\mathcal{F}$ represents the 2-D Fourier transform operator, $\mathbf{B}$ represents the observed "k-space" data [60], and σ represents the variance of the signal noise. It was shown in [120] that using a Bregman iteration technique, problem (2.21) can be reduced to a sequence of unconstrained problems that can be solved using the Split Bregman technique [60]. It is important to note that unlike other formulations presented in the literature, in this chapter problem (2.21) is addressed to deal with images by regarding them as matrix variables instead of column-stacked vectors. Such formulation implicitly applies the separable sensing operator [95, 102] which facilitates efficient analysis, reduces storage complexity, and makes fast computation possible. The matrix-based analysis of TV regularization model is the author's main contribution in this chapter.

In Chapter 5, it is demonstrated that the standard TV can be generalized to a pth-power TV with $0 \leq p \leq 1$, and the generalized TV (GTV) regularized least squares problem produces improved denoising performance relative to the classical methods in the literature. In this chapter, the GTV regularizer is applied to the Fourier-based MRI reconstruction problem, as a result we have to deal with a nonconvex model. There are existing algorithms for nonconvex compressive imaging [36, 37] where the ℓ_1 norm is replaced by the ℓ_p quasi-norm. Unlike the ℓ_1 or ℓ_p regularization, the algorithm proposed in this chapter solves GTV-regularized optimization problem that turns out to perform better than existing algorithms in preserving image edges.

6.1 TV, Generalized TV and Weighted TV with Matrix Representations

The total variation (TV) of an image $\mathbf{U}$ is defined and discussed in Sec. 2.6. In particular, the discretized anisotropic and isotropic TV are defined in (2.17) and (2.18), respectively. Since both TVs are found to yield similar reconstruction results with the performance often slightly in favor of the anisotropic one (see Sec. 5.3), our analysis will be carried out using anisotropic type of TV.

In this section, we shall demonstrate that the TV in (2.17) can be expressed with simple matrix operations. Suppose $\mathbf{U} \in R^{n \times n}$, we define $\mathbf{D} \in R^{n \times n}$ as a circulant matrix with the first row $[1\ -1\ 0\ \cdots\ 0]$. Under the periodic boundary condition [87], it can be verified that the TV can be expressed as

$$\mathrm{TV}(\mathbf{U}) = \|\mathbf{D}\mathbf{U}\|_1 + \|\mathbf{U}\mathbf{D}^T\|_1 \tag{6.2}$$

where $\|\mathbf{X}\|_1$ denotes the sum of magnitudes of all the entries in $\mathbf{X}$, i.e., $\sum |x_{i,j}|$. It follows that the generalized pth power total variation, denoted as TV_p (see Eq. (5.1)), can be written as

$$\mathrm{TV}_p(\mathbf{U}) = \|\mathbf{D}\mathbf{U}\|_p + \|\mathbf{U}\mathbf{D}^T\|_p$$

with $0 \leq p \leq 1$. Note that notation $\|\mathbf{X}\|_p$ resembles but slightly differs from an ℓ_p norm. Specifically, it expresses the sum of pth power magnitudes of all the entries in $\mathbf{X}$, i.e., $\sum |x_{i,j}|^p$. The performance improvement using an ℓ_p norm over the ℓ_1 norm established in [35, 36, 114, 115] inspired us to investigate the generalized total variation, TV_p, for compressive sensing image recovery.

The problem considered in this chapter can be cast as

$$\begin{aligned} &\underset{\mathbf{U}}{\text{minimize}} && \mathrm{TV}_p(\mathbf{U}) && \text{(6.3a)} \\ &\text{subject to:} && \|\mathbf{R} \circ (\mathcal{F}\mathbf{U}) - \mathbf{B}\|_{\mathrm{F}}^2 \leq \sigma^2 && \text{(6.3b)} \end{aligned}$$

The regularizer TV_p is known for promoting a sparser TV when p is less than one. However, TV_p related problem is nonconvex in nature. The idea of reweighting was proposed in Sec. 5.1.2 where we introduced the weighted TV (WTV). In the following, we reformulate WTV of image $\mathbf{U}$ with a matrix representation as

$$\mathrm{TV}_{\mathrm{w}}(\mathbf{U}) = \|\mathbf{W}_x \circ (\mathbf{D}\mathbf{U})\|_1 + \|\mathbf{W}_y \circ (\mathbf{U}\mathbf{D}^T)\|_1 \tag{6.4}$$

where $\circ$ is the Hadamard product operator, and $\mathbf{W}_x$ and $\mathbf{W}_y$ are weights matrices along the horizontal and vertical direction respectively. Clearly, $\mathrm{TV}_{\mathrm{w}}(\mathbf{U})$ becomes the conventional $\mathrm{TV}(\mathbf{U})$ when all entries in $\mathbf{W}_x$ and $\mathbf{W}_y$ are equal to unity.

6.2 A Power-Iterative Reweighting Strategy for Problem (6.3)

Theoretically, solving (6.3) with $p = 0$ promotes the solution with the sparsest TV. In order to approach the solution for a TV_0-regularized optimization problem, we adopt a power-iterative strategy [115] by gradually reducing the power p, updating the weights, and solving a WTV-regularized problem at each iteration. The power-iterative strategy not only properly updates the weights to approach the corresponding TV_p minimization, but also provides the convex WTV-regularized problem with a good initial state based on the previous round of iteration. As a result, the present WTV-regularized problem is found to converge considerably faster as long as the initial state is closer to the globally optimal solution.

In what follows, we denote by $\mathbf{1}$ the matrix with each of its element one, by $|\mathbf{X}|.^p$ a matrix whose (i,j)th entry is $|x_{i,j}|^p$, and by $\circ/$ the pointwise division. The steps of the power-iterative strategy are summarized in Algorithm 6.1.

Algorithm 6.1 Power-Iterative Strategy for TV_p Minimization (6.3)

```
1: Set $p = 1$, $l = 1$, $\mathbf{W}_x = \mathbf{W}_y = \mathbf{1}$.
2: Solve the WTV-regularized problem for $\mathbf{U}^{(l)}$
```

$$\underset{\mathbf{U}}{\text{minimize}} \quad \text{TV}_{\text{w}}(\mathbf{U}) \tag{6.5a}$$

$$\text{subject to:} \quad \|\mathbf{R} \circ (\mathcal{F}\mathbf{U}) - \mathbf{B}\|_{\text{F}}^2 \le \sigma^2 \tag{6.5b}$$

```
3: Terminate if $p = 0$; otherwise, set $p = p - 0.1$ and update the weights $\mathbf{W}_x$ and $\mathbf{W}_y$ as
```

$$\mathbf{W}_x = |\mathbf{D}\mathbf{U}^{(l)} + \epsilon|.^{p-1}, \;\; \mathbf{W}_y = |\mathbf{U}^{(l)}\mathbf{D}^T + \epsilon|.^{p-1} \tag{6.6}$$

```
   Then set $l = l + 1$ and repeat from step 2.
```

By observing Algorithm 6.1, we remark that $\text{TV}_{\text{w}}(\mathbf{U})$ essentially becomes $\text{TV}_p(\mathbf{U})$ for $\mathbf{U}$ in a neighborhood of iterate $\mathbf{U}^{(l)}$, by the reweighting formula (6.6) (refer to (5.7)). The parameter ϵ in (6.6) is a small constant to prevent the weights from being zero. In this way, nonconvex minimization of $\text{TV}_p(\mathbf{U})$ can practically be achieved by a series of convex minimization of $\text{TV}_{\text{w}}(\mathbf{U})$ using the above power-iterative strategy.

6.3 WTV-Regularized Minimization for Problem (6.5)

The analysis has led to a WTV-regularized problem as seen in (6.5). We propose to solve the problem using a Split Bregman [60] approach, but with important changes as will be described in the following. Note that the Split Bregman method is found to be equivalent to the augmented Lagrangian method in the context of total variation minimization [55, 103]. Unlike vector operations in the literature, the entire analysis presented below is carried out in terms of matrix operations.

6.3.1 Split Bregman Type Iteration

First let us apply the Bregman iteration [120] to (6.5) which reduces the problem to

$$\mathbf{U}^{(k+1)} = \underset{\mathbf{U}}{\operatorname{argmin}}\, \mathrm{TV}_{\mathrm{w}}(\mathbf{U}) + \frac{\mu}{2}\|\mathbf{R}\circ(\mathcal{F}\mathbf{U}) - \mathbf{B}^{(k)}\|_{\mathrm{F}}^2 \tag{6.7a}$$

$$\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)} + \mathbf{B} - \mathbf{R}\circ(\mathcal{F}\mathbf{U}^{(k+1)}) \tag{6.7b}$$

With Eq. (6.4), problem (6.7) can be expressed as

$$\mathbf{U}^{(k+1)} = \underset{\mathbf{U}}{\operatorname{argmin}}\, \|\mathbf{W}_x\circ(\mathbf{D}\mathbf{U})\|_1 + \|\mathbf{W}_y\circ(\mathbf{U}\mathbf{D}^T)\|_1 + \frac{\mu}{2}\|\mathbf{R}\circ(\mathcal{F}\mathbf{U}) - \mathbf{B}^{(k)}\|_{\mathrm{F}}^2 \tag{6.8a}$$

$$\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)} + \mathbf{B} - \mathbf{R}\circ(\mathcal{F}\mathbf{U}^{(k+1)}) \tag{6.8b}$$

Next, a splitting strategy applied to (6.8a) leads to the formulation

$$\begin{aligned} \underset{\mathbf{U}}{\text{minimize}} \quad & \|\mathbf{W}_x\circ\mathbf{D}_x\|_1 + \|\mathbf{W}_y\circ\mathbf{D}_y\|_1 + \frac{\mu}{2}\|\mathbf{R}\circ(\mathcal{F}\mathbf{U}) - \mathbf{B}^{(k)}\|_{\mathrm{F}}^2 & \text{(6.9a)} \\ \text{subject to:} \quad & \mathbf{D}_x = \mathbf{D}\mathbf{V}, \mathbf{D}_y = \mathbf{V}\mathbf{D}^T, \mathbf{U} = \mathbf{V} & \text{(6.9b)} \end{aligned}$$

Note that in (6.9) we have applied the Split Bregman technique [60] for the splitting $\mathbf{D}_x = \mathbf{D}\mathbf{V}$ and $\mathbf{D}_y = \mathbf{V}\mathbf{D}^T$, and introduce an additional split as $\mathbf{U} = \mathbf{V}$. The condition $\mathbf{U} = \mathbf{V}$ guarantees that this is, in fact, the same one variable; however such a split allows us to decompose the most computationally expensive step of the algorithm into two much simpler steps [102], as will be demonstrated in the following.

Applying Bregman method again to (6.9) to enforce constraints in (6.9b), we instead minimize the following function with respect to $\{\mathbf{U}, \mathbf{V}, \mathbf{D}_x, \mathbf{D}_y\}$

$$\begin{aligned} \text{minimize}\ & \|\mathbf{W}_x\circ\mathbf{D}_x\|_1 + \|\mathbf{W}_y\circ\mathbf{D}_y\|_1 + \frac{\mu}{2}\|\mathbf{R}\circ(\mathcal{F}\mathbf{U}) - \mathbf{B}^{(k)}\|_{\mathrm{F}}^2 \\ & + \frac{\lambda}{2}\|\mathbf{D}_x - \mathbf{D}\mathbf{V} - \mathbf{E}_x^{(h)}\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\|\mathbf{D}_y - \mathbf{V}\mathbf{D}^T - \mathbf{E}_y^{(h)}\|_{\mathrm{F}}^2 \\ & + \frac{\nu}{2}\|\mathbf{U} - \mathbf{V} - \mathbf{G}^{(h)}\|_{\mathrm{F}}^2 \end{aligned}$$

where $\mathbf{E}_x^{(h)}$, $\mathbf{E}_y^{(h)}$ and $\mathbf{G}^{(h)}$ are updated through Bregman iterations. In the hth iteration, we solve four subproblems to obtain $\mathbf{U}^{(h+1)}$, $\mathbf{V}^{(h+1)}$, $\mathbf{D}_x^{(h+1)}$ and $\mathbf{D}_y^{(h+1)}$ as

$$\mathbf{U}^{(h+1)} = \underset{\mathbf{U}}{\text{argmin}} \ \frac{\mu}{2}\|\mathbf{R} \circ (\mathcal{F}\mathbf{U}) - \mathbf{B}^{(k)}\|_\mathrm{F}^2 + \frac{\nu}{2}\|\mathbf{U} - \mathbf{V}^{(h)} - \mathbf{G}^{(h)}\|_\mathrm{F}^2 \tag{6.10}$$

$$\begin{aligned}\mathbf{V}^{(h+1)} = \underset{\mathbf{V}}{\text{argmin}} \ & \frac{\nu}{2}\|\mathbf{V} - \mathbf{U}^{(h+1)} + \mathbf{G}^{(h)}\|_\mathrm{F}^2 \\ &+ \frac{\lambda}{2}\|\mathbf{D}\mathbf{V} + \mathbf{E}_x^{(h)} - \mathbf{D}_x^{(h)}\|_\mathrm{F}^2 + \frac{\lambda}{2}\|\mathbf{V}\mathbf{D}^T + \mathbf{E}_y^{(h)} - \mathbf{D}_y^{(h)}\|_\mathrm{F}^2\end{aligned} \tag{6.11}$$

$$\mathbf{D}_x^{(h+1)} = \underset{\mathbf{D}_x}{\text{argmin}} \ \|\mathbf{W}_x \circ \mathbf{D}_x\|_1 + \frac{\lambda}{2}\|\mathbf{D}_x - \mathbf{D}\mathbf{V}^{(h+1)} - \mathbf{E}_x^{(h)}\|_\mathrm{F}^2 \tag{6.12a}$$

$$\mathbf{D}_y^{(h+1)} = \underset{\mathbf{D}_y}{\text{argmin}} \ \|\mathbf{W}_y \circ \mathbf{D}_y\|_1 + \frac{\lambda}{2}\|\mathbf{D}_y - \mathbf{V}^{(h+1)}\mathbf{D}^T - \mathbf{E}_y^{(h)}\|_\mathrm{F}^2 \tag{6.12b}$$

where the iterates $\mathbf{G}^{(h)}$, $\mathbf{E}_x^{(h)}$ and $\mathbf{E}_y^{(h)}$ are updated by Bregman iteration as

$$\mathbf{G}^{(h+1)} = \mathbf{G}^{(h)} + \mathbf{V}^{(h+1)} - \mathbf{U}^{(h+1)} \tag{6.13a}$$

$$\mathbf{E}_x^{(h+1)} = \mathbf{E}_x^{(h)} + \mathbf{D}\mathbf{V}^{(h+1)} - \mathbf{D}_x^{(h+1)} \tag{6.13b}$$

$$\mathbf{E}_y^{(h+1)} = \mathbf{E}_y^{(h)} + \mathbf{V}^{(h+1)}\mathbf{D}^T - \mathbf{D}_y^{(h+1)} \tag{6.13c}$$

Typically, it takes a small number of iterations of (6.10), (6.11), (6.12) and (6.13) for the algorithm to converge to the minimizer of (6.7a) as mandated by the theory of Bregman iterations.

Note that the problems in (6.12a) and (6.12b) can be solved by soft shrinkage as the unknowns are separate from each other. Specifically,

$$\mathbf{D}_x^{(h+1)} = \mathcal{T}_{\mathbf{W}_x/\lambda}(\mathbf{D}\mathbf{V}^{(h+1)} + \mathbf{E}_x^{(h)}) \tag{6.14a}$$

$$\mathbf{D}_y^{(h+1)} = \mathcal{T}_{\mathbf{W}_y/\lambda}(\mathbf{V}^{(h+1)}\mathbf{D}^T + \mathbf{E}_y^{(h)}) \tag{6.14b}$$

where soft shrinkage operator $\mathcal{T}$ applies pointwisely as

$$\mathcal{T}_{w_{i,j}/\lambda}(z) = \text{sgn}(z) \cdot \max\{|z| - w_{i,j}/\lambda, 0\} \tag{6.15}$$

6.3.2 Solving Problems (6.10) and (6.11)

Solving the remaining problems (6.10) and (6.11) are however far from trivial. To this end, we first write first-order optimality condition of (6.10) as

$$\mu \mathcal{F}^T \mathbf{R} \circ \mathcal{F}\mathbf{U} + \nu \mathbf{U} = \mu \mathcal{F}^T \mathbf{R} \circ \mathbf{B}^k + \nu(\mathbf{V}^{(h)} + \mathbf{G}^{(h)}) \tag{6.16}$$

Multiplying both sides of (6.16) by $\mathcal{F}$ on the left and applying the orthogonality of Fourier transform, i.e., $\mathcal{F}^T\mathcal{F} = \mathcal{I}$, we have

$$\mu \mathbf{R} \circ \mathcal{F}\mathbf{U} + \nu \mathcal{F}\mathbf{U} = \mu \mathbf{R} \circ \mathbf{B}^k + \nu \mathcal{F}(\mathbf{V}^{(h)} + \mathbf{G}^{(h)}) \tag{6.17}$$

Furthermore, the Fourier transform of $\mathbf{U}$ can be derived as

$$\mathcal{F}\mathbf{U} = \left[\mu \mathbf{R} \circ \mathbf{B}^k + \nu \mathcal{F}(\mathbf{V}^{(h)} + \mathbf{G}^{(h)})\right] \circ / \ (\mu \mathbf{R} + \nu) \tag{6.18}$$

Finally, by taking the inverse Fourier transform on both sides of (6.18), the solution of problem (6.10) has the following representation

$$\mathbf{U}^{(h+1)} = \mathcal{F}^T \left\{ \left[\mu \mathbf{R} \circ \mathbf{B}^k + \nu \mathcal{F}(\mathbf{V}^{(h)} + \mathbf{G}^{(h)})\right] \circ / \ (\mu \mathbf{R} + \nu) \right\} \tag{6.19}$$

Solving problem (6.11) is more involved. First, using matrix calculus [62] we write its first-order optimality condition as

$$\nu \mathbf{V} + \lambda \mathbf{D}^T \mathbf{D} \mathbf{V} + \lambda \mathbf{V} \mathbf{D}^T \mathbf{D} = \mathbf{C}^{(h)} \tag{6.20}$$

where

$$\mathbf{C}^{(h)} = \nu(\mathbf{U}^{(h+1)} - \mathbf{G}^{(h)}) + \lambda \mathbf{D}^T(\mathbf{D}_x^{(h)} - \mathbf{E}_x^{(h)}) + \lambda(\mathbf{D}_y^{(h)} - \mathbf{E}_y^{(h)})\mathbf{D} \tag{6.21}$$

Since the periodic boundary condition is used, matrix $\mathbf{D}$ is circulant and can be diagonalized by the 2-D Fourier transform as [87, 121]

$$\mathbf{D} = \mathcal{F}^T \mathbf{\Lambda} \mathcal{F} \tag{6.22}$$

where $\mathbf{\Lambda}$ is a diagonal matrix. By substitution of (6.22) into (6.20), we have

$$\nu \mathbf{V} + \lambda \mathcal{F}^T \mathbf{\Lambda}^* \mathbf{\Lambda} \mathcal{F} \mathbf{V} + \lambda \mathbf{V} \mathcal{F}^T \mathbf{\Lambda}^* \mathbf{\Lambda} \mathcal{F} = \mathbf{C}^{(h)} \tag{6.23}$$

which can be further reduced to the following by multiplying both sides by $\mathcal{F}$ on the left and $\mathcal{F}^T$ on the right

$$\nu\tilde{\mathbf{V}} + \lambda(\mathbf{T}\tilde{\mathbf{V}} + \tilde{\mathbf{V}}\mathbf{T}) = \mathcal{F}\mathbf{C}^{(h)}\mathcal{F}^T \tag{6.24}$$

where $\tilde{\mathbf{V}} = \mathcal{F}\mathbf{V}\mathcal{F}^T$ and $\mathbf{T} = \mathbf{\Lambda}^*\mathbf{\Lambda}$. Since $\mathbf{T}$ is also a diagonal matrix, it can be verified that

$$\mathbf{T}\tilde{\mathbf{V}} = \mathbf{T}_r \circ \tilde{\mathbf{V}},\ \ \tilde{\mathbf{V}}\mathbf{T} = \mathbf{T}_c \circ \tilde{\mathbf{V}}$$

where $\mathbf{T}_r$ has each element in its ith row as $T_{i,i}$ and $\mathbf{T}_c$ has each element in its ith column as $T_{i,i}$. Thus, Eq. (6.24) can be expressed as

$$(\nu + \lambda\mathbf{T}_r + \lambda\mathbf{T}_c) \circ \tilde{\mathbf{V}} = \mathcal{F}\mathbf{C}^{(h)}\mathcal{F}^T \tag{6.25}$$

In consequence, we obtain solution of (6.11) as

$$\mathbf{V}^{(h+1)} = \mathcal{F}^T \left\{(\mathcal{F}\mathbf{C}^{(h)}\mathcal{F}^T) \circ / \ (\nu + \lambda\mathbf{T}_r + \lambda\mathbf{T}_c)\right\} \mathcal{F} \tag{6.26}$$

We now summarize the algorithm for solving the WTV-regularized problem (6.5) as Algorithm 6.2.

Algorithm 6.2 Algorithm for WTV-regularized problem (6.5)

```
Set $\mu$, $\lambda$ and $\nu$. Set maximum inner and outer iteration number $K$ and $H$.
Initialize $\mathbf{B}^{(0)}$, $\mathbf{V}^{(0)}$, $\mathbf{G}^{(0)}$, $\mathbf{D}_x^{(0)}$, $\mathbf{D}_y^{(0)}$, $\mathbf{E}_x^{(0)}$ and $\mathbf{E}_y^{(0)}$.
for $k = 0, ..., K-1$ do
    for $h = 0, ..., H-1$ do
        Compute $\mathbf{U}^{(h+1)}$, $\mathbf{V}^{(h+1)}$, $\mathbf{D}_x^{(h+1)}$, $\mathbf{D}_y^{(h+1)}$ by Eqs. (6.19), (6.26) and (6.14), respectively. Update $\mathbf{G}^{(h+1)}$, $\mathbf{E}_x^{(h+1)}$, $\mathbf{E}_y^{(h+1)}$ through Eq. (6.13).
    end for
    Set $\mathbf{U}^{(k+1)} = \mathbf{U}^{(H)}$. Update $\mathbf{B}^{(k+1)}$ by Eq (6.7b).
end for
```

Since the computational steps involved in Algorithm 6.2 are linear operations, we remark that the time and space complexity of the algorithm also scales linearly to the size of the image. In the following section, we demonstrate the performance of proposed Algorithms 6.1 and 6.2 in image reconstruction from compressive samples on a series of synthetic and natural images of size 256 by 256.

6.4 Performance on Compressive Imaging

6.4.1 MRI of the Shepp-Logan Phantom

The Shepp-Logan phantom is a standard test image created by Larry Shepp and Benjamin F. Logan [101]. It serves as the model of a human head in the development and testing of image reconstruction algorithms [70, 82], and is used widely by researchers in tomography. An original 256 × 256 Shepp-Logan phantom is illustrated in Fig. 6.1.

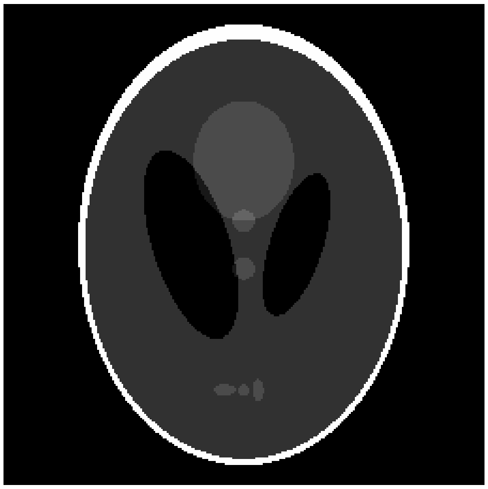


Figure 6.1: A Shepp-Logan phantom

In this experiment, a normalized Shepp-Logan phantom of size 256 × 256, was measured at 2521 locations (as low as 3.85%) in the 2D Fourier plane (k-space); the sampling pattern was a star-shaped pattern consisting of only 10 radial lines, see Fig. 6.2(a). The ℓ_1-magic toolbox [25] was utilized to create the star-shaped sampling pattern. Based on the 2521 star-shaped 2D Fourier samples, a minimum ℓ_2 norm reconstruction result is shown in Fig. 6.2(b).

We carried out the power-iterative strategy for GTV minimization by Algorithm 6.1 with implementation in MATLAB. Initially, we set $p = 1$ and $\mathbf{W}_x$ and $\mathbf{W}_y$ as all one matrices. In each round of iteration, the WTV-regularized problem (6.5) was solved by Algorithm 6.2. Parameters μ, λ and ν were all set to 5. We remark that these parameters were chosen arbitrarily and the problem considered was tolerant to the values to a great extent by virtue of the Split Bregman algorithm. The inner and outer iterations were set to $H = 10$ and $K = 100$, where the number of iterations were chosen to be more than sufficient for convergence of the algorithm. It is worthwhile

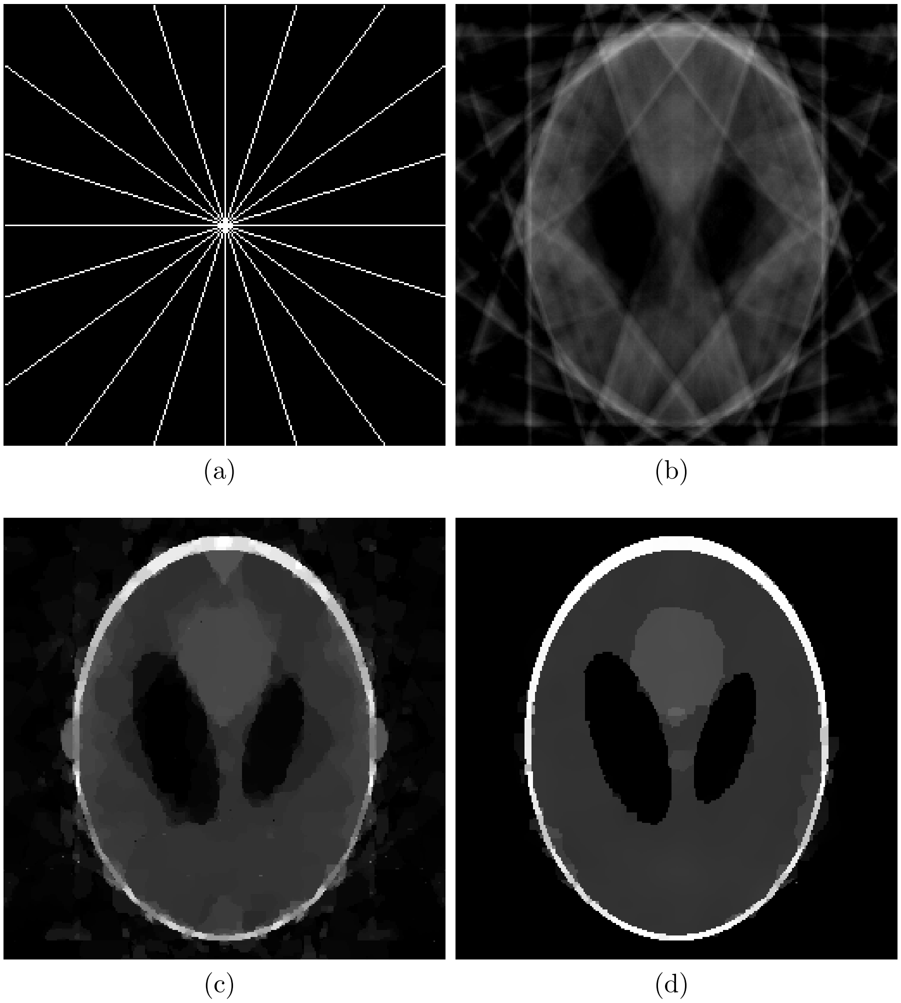


Figure 6.2: (a) Star-shaped sampling pattern (b) Minimum energy reconstruction (c) Minimum TV reconstruction (d) Minimum GTV reconstruction with $p = 0$

to remark that other than presetting the maximum number of iterations, convergence criteria of the algorithm can also be prescribed as the magnitude difference between two adjacent iterates becoming less than a certain tolerance. In this way, similar level of reconstruction quality can be achieved depending on the pre-defined tolerance value. The initial values for $\mathbf{B}^{(0)}$, $\mathbf{V}^{(0)}$, $\mathbf{G}^{(0)}$, $\mathbf{D}_x^{(0)}$, $\mathbf{D}_y^{(0)}$, $\mathbf{E}_x^{(0)}$ and $\mathbf{E}_y^{(0)}$ were all set to zero matrices. The minimizer of problem (6.5) was then used to update $\mathbf{W}_x$ and $\mathbf{W}_y$ for the next round when $p = 0.9$ in Algorithm 6.1. It is important to remark that starting from the 2nd round in solving problem (6.5), $\mathbf{B}^{(0)}$, $\mathbf{V}^{(0)}$, $\mathbf{G}^{(0)}$, $\mathbf{D}_x^{(0)}$, $\mathbf{D}_y^{(0)}$, $\mathbf{E}_x^{(0)}$ and $\mathbf{E}_y^{(0)}$ were initialized using the most recent iterates from the last cycle. Such an initialization ensures problem (6.5) start with an initial point not far away from the global minimizer, reducing the iteration number for convergence as a result. The steps were carried on for $p = 0.8, 0.7, 0.6, ...$ and so forth until the result for $p = 0$ is achieved.

It took a PC laptop with a 2.67 GHz Intel quad-core processor 770.7 seconds to produce the reconstructed phantom for $p = 0$ shown in Fig. 6.2(d). The signal to noise (SNR) ratio was found to be 16.3 dB. For a fair comparison, we set $\mathbf{W}_x = \mathbf{W}_y = \mathbf{1}$, and minimize problem (6.5) by Algorithm 6.2 with the outer iteration number K set as 1100. The computational time was found to be 756.8s. The solution simply corresponds to the conventional TV minimization recovery (2.21). The maximum SNR conventional TV minimization can achieve was found to be 8.8 dB, as illustrated in Fig. 6.2(c). Therefore, using the proposed method to approach the TV_0 solution, we have observed better reconstruction performance relative to the conventional TV minimization.

6.4.2 Compressive Imaging of Natural Images

To further examine the proposed GTV minimization algorithm, we extend the simulation to compressive sensing of several natural images - `cameraman`, `building`, `milk` and `jet`. Each test image of size 256 by 256 was measured at 13107 random locations (i.e., 20% of size of image) in the 2D Fourier plane. The random sampling pattern $\mathbf{R}$ was shown as a black and white image in Fig. 6.3, where a dark pixel represents 0, and a white pixel represents 1.

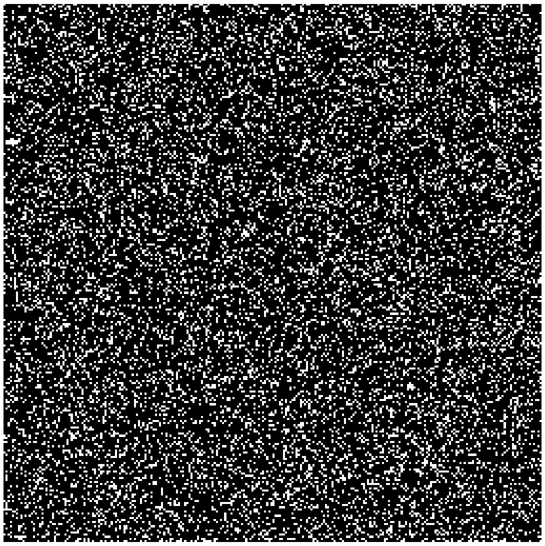
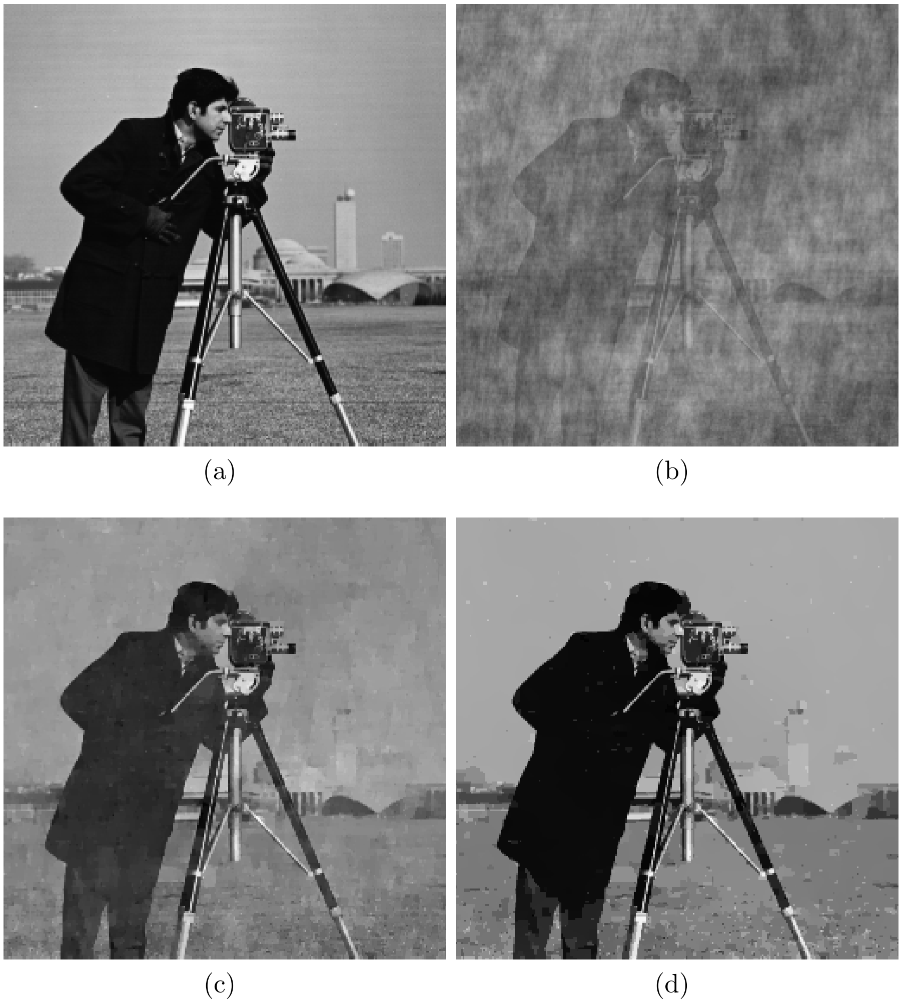


Figure 6.3: Random sampling pattern

The parameter setting was the same as in Sec. 6.4.1. The power iterative technique was also applied to reduce the power p from 1 to 0 with a 0.1 step each time, in order to ensure a decent initial point. Reconstruction performance of the proposed GTV ($p = 0$) algorithm compared with ℓ_2 and TV based minimization were illustrated in Figs. 6.4-6.7.

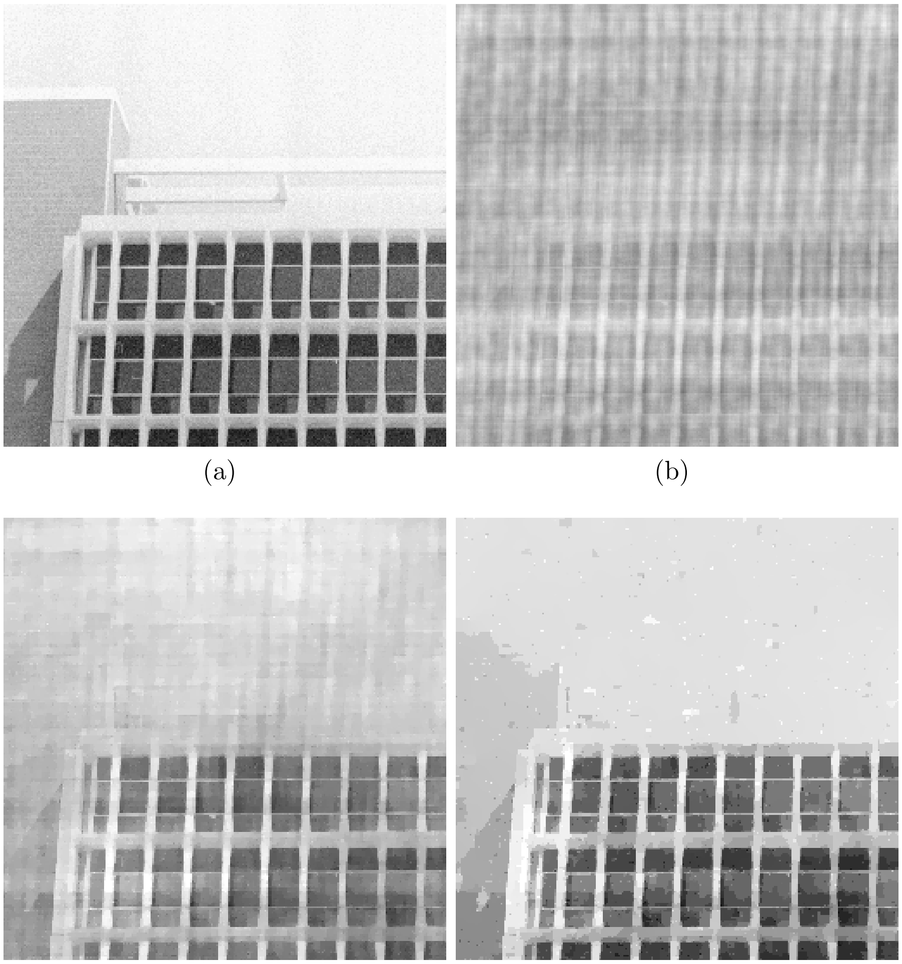
It was found that the images reconstructed using the proposed GTV minimization method possess consistently higher SNRs than those from the conventional minimum TV reconstruction. Visual inspection of Figs. 6.4-6.7 further demonstrates that the proposed GTV minimization algorithm has an edge on the conventional TV model in reconstructing images from compressive measurements.



(a) (b)

(c) (d)

Figure 6.4: (a) Image `cameraman` (b) Minimum energy reconstruction (c) Minimum TV reconstruction (SNR = 14.3 dB) (d) Minimum GTV reconstruction with $p = 0$ (SNR = 19.5 dB)



(a)

(b)

(c)

(d)

Figure 6.5: (a) Image `building` (b) Minimum energy reconstruction (c) Minimum TV reconstruction (SNR = 15.2 dB) (d) Minimum GTV reconstruction with $p = 0$ (SNR = 18.3 dB)

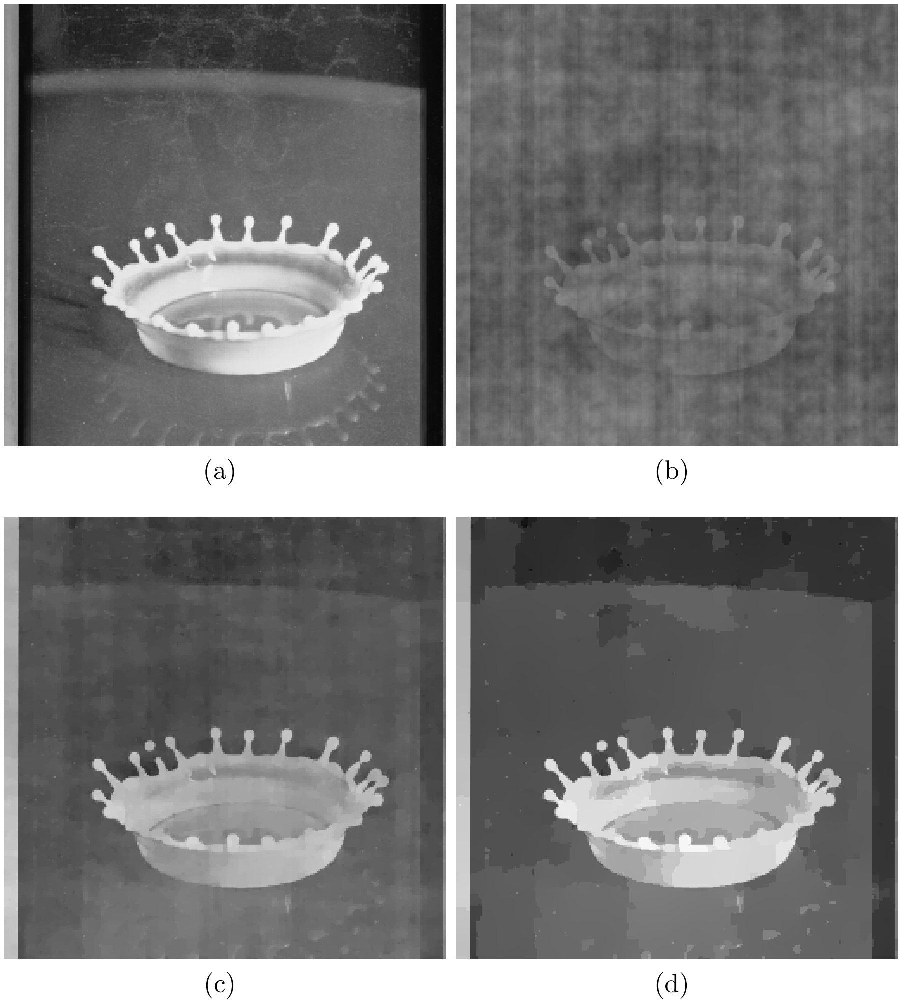
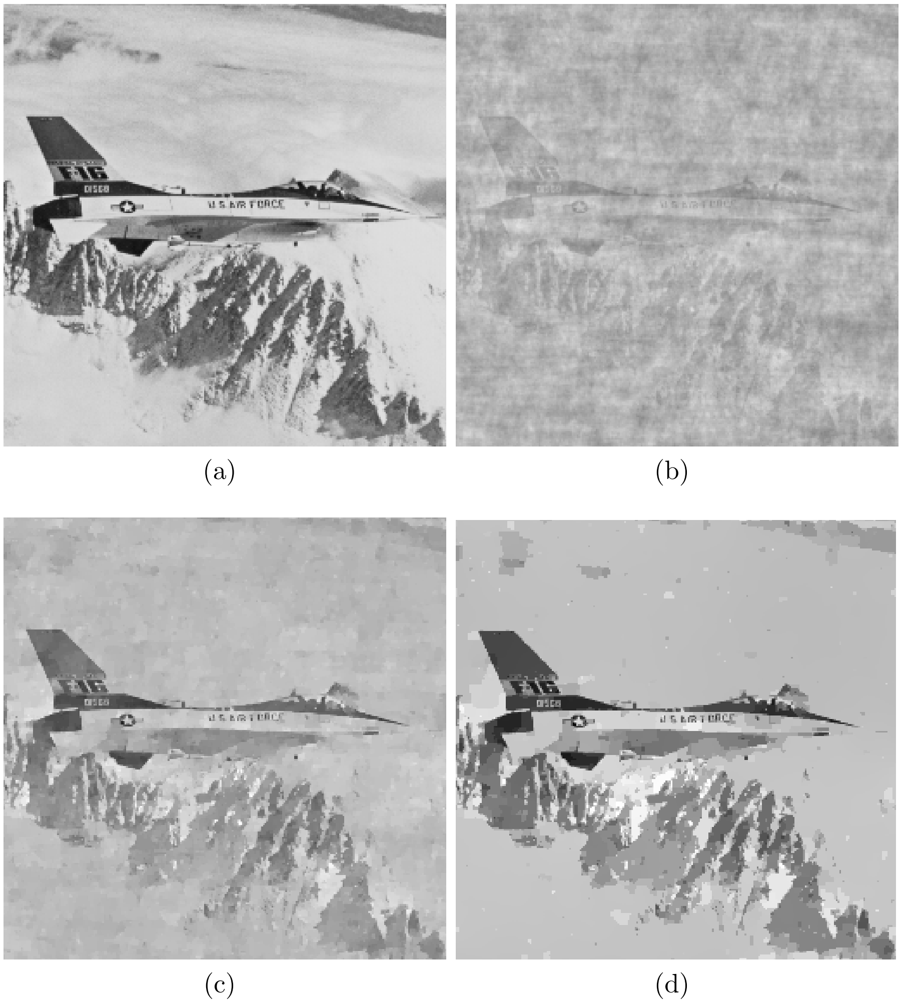


Figure 6.6: (a) Image `milk` (b) Minimum energy reconstruction (c) Minimum TV reconstruction (SNR = 12.1 dB) (d) Minimum GTV reconstruction with $p = 0$ (SNR = 14.5 dB)



(a)

(b)

(c)

(d)

Figure 6.7: (a) Image `jet` (b) Minimum energy reconstruction (c) Minimum TV reconstruction (SNR = 16.2 dB) (d) Minimum GTV reconstruction with $p = 0$ (SNR = 18.3 dB)

Chapter 7

Concluding Remarks

This thesis investigates several new techniques for ℓ_1-regularized problem, TV-regularized problem, and their nonconvex relaxations. The algorithmic issues and performance of the proposed methods have been investigated and applied to the general area of signal reconstruction, including signal and image denoising, signal sparse representation, compressive sensing and compressive imaging. The objectives of this thesis are two-fold. Firstly, by extending models from convex to nonconvex, several methods have been analyzed to approach globally optimal solution and to improve quality of reconstructed signal (in Chapters 3, 5, 6). Secondly, the thesis has also addressed practical application of the mathematical models by developing a solver for parallel processing, designing accelerated algorithms with faster convergence rate, and presenting a matrix-based analysis for convenient implementation and coding (in Chapters 3, 4, 6).

In Chapter 3, a power-iterative strategy has been proposed for compressive sensing in an ℓ_p-ℓ_2 minimization setting. The methodology is built on a modified FISTA developed for local solution of the ℓ_p-ℓ_2 problem, in which a parallel global solver is devised for the proximal-point function. Experimental results are presented to show the superiority of the algorithms compared with the conventional BP benchmarks, and to demonstrate that the solutions obtained are highly likely to be globally optimal. In addition, a smoothed ℓ_p-ℓ_2 solver for signal spaces with orthogonal basis or overcomplete dictionary have been proposed. The solver is computationally efficient because the solver with orthogonal basis is non-iterative while the solver with overcomplete dictionary admits FISTA type iterations for fast convergence. The proposed solver is demonstrated to outperform its ℓ_1-ℓ_2 counterpart for signal denoising.

In Chapter 4, a fast dual-based linearized Bregman algorithm has been proposed for the equality constrained nonsmooth convex programming. The algorithm is carried out for a dual problem, making the selection and adjustment of the regularization parameter rather straightforward. The algorithm's acceleration is made possible by enhancing each gradient descent iteration in a way similar to that employed in FISTA. Performance and complexity of the fast algorithm are evaluated and compared with the conventional LB algorithm by applying to CS reconstruction of 1-D sparse signals. In addition, performance of the proposed algorithm in dealing with large-scale data is demonstrated by accurately reconstructing several test images.

We stress that matrix $\mathbf{A} = \mathbf{C\Psi}$ was adopted as the measurement matrix for obtaining the compressive sampled data $\mathbf{b} = \mathbf{Ax}^*$. Nevertheless, if we were to design a compressive digital camera, matrix $\mathbf{A}$ would be implemented as a digital micromirror device (DMD) of an array of N tiny mirrors for light reflection that further focuses onto a single photodiode (the single pixel). The process would be repeated M times, where at the kth time, the array of N tiny mirrors corresponds to instantiation of the kth row of $\mathbf{A}$. The reader is referred to [6, 7] for more details with regard to the practicality of the "single-pixel" CS camera. Compared to a $0/1$ random matrix, implementation of $\mathbf{A} = \mathbf{C\Psi}$ as arrays of tiny mirrors is costly in both computation and storage. Therefore, measurement matrix with simpler structure yet guaranteeing exact recovery performance is one major element in CS from a hardware implementation point of view. A notable contribution has been made by Yin, Morgan, Yang and Zhang [121], who discovered that optimal incoherence can be achieved by random Toeplitz and circulant matrices. Furthermore, [113] learns a circulant matrix from training data and demonstrates that the learned matrix outperforms random ones. It appears worthwhile to explore hardware implementation of circulant matrix for obtaining compressive sampled data, as well as fast computational methods with prior knowledge of such measurement matrices for practical compressive imaging.

The concept of TV has been generalized to a pth power TV in Chapter 5. Due to the nonconvex nature of the TV$_p$-regularized problem, we deal with the image denoising problem by proposing a weighted TV minimization where the weights are updated iteratively to solve the problem in a convex-programming setting. The technical difficulties of WTV minimization are addressed in a modified Split Bregman framework. Numerical results have indicated that, with an appropriate power $p < 1$, the proposed IRTV algorithm enhances denoising performance relative to several recent denoising algorithms from the literature.

Chapter 6 presents an algorithm for the reconstruction of digital images from undersampled measurements, where the concept of generalized TV (GTV) that involves pth power of the discretized gradient of the image is utilized. To deal with the nonconvex issue arising from this new formulation, a weighted TV-regularized problem has been solved in the Split Bregman framework with additional splitting technique. The algorithm adopts a power-iterative strategy that gradually reduces the power p from 1 to 0. In particular, the MRI problem considered in this chapter is addressed by regarding image variables as matrices rather than column-stacked vectors. Numerical simulations have been performed using a variety of medical and natural images. The proposed technique is found to be superior relative to the conventional TV minimization method in terms of the quality of the reconstructed images.

The thesis has been focused on denoising and compressive sensing. Nevertheless, the concept of ℓ_p or TV$_p$-regularization and techniques developed here can be extended to a broad range of applications including signal deconvolution, image deblurring, and other related problems. In the literature, linearized Bregman method is used to solve the matrix completion problem in [16], as well as image deblurring in [17, 18, 20]. Split Bregman method is another popular building block for solving optimization model involving TV, which has been applied to image segmentation problems [61] and to estimate nonsmooth probability densities [81]. It appears to be worthwhile to investigate broader applications of sparse optimization problems involving ℓ_p, TV$_p$, or a combination of those regularization terms. Very recent work considers sparse optimization in a parallel and distributed manner that closely mimics the computational environment nowadays. Several methods have been proposed to deal with very large-scale basis pursuit problem [43], distributed LASSO, sparse logistic regression [94], and study of decentralized gradient descent for consensus optimization problems in multi-agent networks [122]. Because of the complexity involved and generally large-scale nature of these problems, faster and easily manageable algorithmic and software solutions are vital for real-world applications where a tradeoff normally needs to be attained in consideration of time, space, and cost requirements.

Bibliography

[1] M. Adams, *Wavelet Systems and Multirate Filter Banks: Theory and Practice - 2010 edition*. ELEC 486/586 course Notes, University of Victoria, 2010.

[2] W. K. Allard, "Total variation regularization for image denoising, II. Examples," *SIAM J. Imaging Sciences*, vol. 1, no. 4, pp. 400–417, 2008.

[3] ——, "Total variation regularization for image denoising, III. Examples," *SIAM J. Imaging Sciences*, vol. 2, no. 2, pp. 532–568, 2009.

[4] A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*. Springer-Verlag New York Inc, 2007.

[5] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.

[6] R. G. Baraniuk, "Compressive sensing," *IEEE signal processing magazine*, vol. 24, no. 4, 2007.

[7] ——, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, 2008.

[8] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[9] ——, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *Image Processing, IEEE Transactions on*, vol. 18, no. 11, pp. 2419–2434, 2009.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[11] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2009.

[12] L. M. Bregman, “The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming,” *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, 1967.

[13] R. Bruck, “On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in Hilbert space,” *Journal of Mathematical Analysis and Applications*, vol. 61, no. 1, pp. 159–164, 1977.

[14] A. Bruckstein, D. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM Review*, vol. 51, no. 1, p. 34, 2009.

[15] J. Cai, S. Osher, and Z. Shen, “Convergence of the linearized Bregman iteration for ℓ_1-norm minimization,” *Mathematics of Computation*, vol. 78, no. 268, pp. 2127–2136, 2009.

[16] J.-F. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[17] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, “Blind motion deblurring from a single image using sparse approximation,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 104–111.

[18] ——, “Blind motion deblurring using multiple images,” *Journal of computational physics*, vol. 228, no. 14, pp. 5057–5071, 2009.

[19] J.-F. Cai, S. Osher, and Z. Shen, “Linearized Bregman iterations for compressed sensing,” *Mathematics of Computation*, vol. 78, no. 267, pp. 1515–1536, 2009.

[20] ——, “Linearized Bregman iterations for frame-based image deblurring,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 226–252, 2009.

[21] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[22] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Info. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[23] E. Candès and M. Wakin, "An introduction to compressive sampling," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 21–30, 2008.

[24] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[25] E. Candès and J. Romberg, "ℓ_1-magic: Recovery of sparse signals via convex programming," *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, vol. 4, 2005.

[26] ——, "Signal recovery from random projections," in *Proc. SPIE*, vol. 5674, 2005, pp. 76–86.

[27] E. J. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, 2006, pp. 1433–1452.

[28] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1, pp. 89–97, 2004.

[29] ——, "Total variation minimization and a class of binary MRF models," in *Energy minimization methods in computer vision and pattern recognition.* Springer, 2005, pp. 136–152.

[30] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numerische Mathematik*, vol. 76, no. 2, pp. 167–188, 1997.

[31] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Recent developments in total variation image restoration," *Mathematical Models of Computer Vision*, vol. 17, 2005.

[32] T. F. Chan and S. Esedoglu, "Aspects of total variation regularized ℓ_1 function approximation," *SIAM Journal on Applied Mathematics*, vol. 65, no. 5, pp. 1817–1837, 2005.

[33] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM Journal on Scientific Computing*, vol. 20, no. 6, pp. 1964–1977, 1999.

[34] T.-C. Chang, L. He, and T. Fang, "MR image reconstruction from sparse radial samples using Bregman iteration," in *Proceedings of the 13th Annual Meeting of ISMRM, Seattle*, 2006, p. 696.

[35] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *Signal Processing Letters, IEEE*, vol. 14, no. 10, pp. 707–710, 2007.

[36] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *ICASSP*, 2008, pp. 3869–3872.

[37] R. Chartrand, "Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data," in *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on.* IEEE, 2009, pp. 262–265.

[38] Q. Chen, P. Montesinos, Q. S. Sun, P. A. Heng, and D. S. Xia, "Adaptive total variation denoising based on difference curvature," *Image and Vision Computing*, vol. 28, no. 3, pp. 298–306, 2010.

[39] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[40] X. Chen and W. Zhou, "Convergence of reweighted ℓ_1 minimization algorithms and unique solution of truncated ℓ_p minimization," *Technical Report*, 2010.

[41] A. Chopra and H. Lian, "Total variation, adaptive total variation and nonconvex smoothly clipped absolute deviation penalty for denoising blocky images," *Pattern Recognition*, vol. 43, no. 8, pp. 2609–2619, 2010.

[42] Y.-H. Dai and R. Fletcher, “Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming,” *Numerische Mathematik*, vol. 100, no. 1, pp. 21–47, 2005.

[43] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, “Parallel multi-block ADMM with O(1/k) convergence,” *UCLA CAM 13-64*, 2013.

[44] D. L. Donoho and X. Huo, “Uncertainty principles and ideal atomic decomposition,” *Information Theory, IEEE Transactions on*, vol. 47, no. 7, pp. 2845–2862, 2001.

[45] D. L. Donoho and P. B. Stark, “Uncertainty principles and signal recovery,” *SIAM Journal on Applied Mathematics*, vol. 49, no. 3, pp. 906–931, 1989.

[46] D. Donoho, “Denoising by soft-thresholding,” *IEEE Trans. Inform. Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[47] ——, “Compressed sensing,” *IEEE Trans. Info. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[48] D. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, p. 2197, 2003.

[49] ——, “On the stability of the basis pursuit in the presence of noise,” *Signal Processing*, vol. 86, no. 3, pp. 511–532, 2006.

[50] D. Donoho and I. Johnstone, “Adapting to unknown smoothness via wavelet shrinkage,” *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[51] D. Donoho and Y. Tsaig, “Fast solution of ℓ_1-norm minimization problems when the solution may be sparse,” *IEEE Trans. Information Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.

[52] M. F. Duarte and R. G. Baraniuk, “Kronecker compressive sensing,” *Image Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 494–504, 2012.

[53] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Annals of statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[54] M. Elad and A. M. Bruckstein, “A generalized uncertainty principle and sparse representation in pairs of bases,” *Information Theory, IEEE Transactions on*, vol. 48, no. 9, pp. 2558–2567, 2002.

[55] E. Esser, “Applications of lagrangian-based alternating direction methods and connections to split bregman,” *CAM report*, vol. 9, p. 31, 2009.

[56] R. Fletcher, “On the Barzilai-Borwein method,” in *Optimization and control with applications.* Springer, 2005, pp. 235–256.

[57] S. Foucart and M. Lai, “Sparsest solutions of underdetermined linear systems via lq-minimization for $0 < q < 1$,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 395–407, 2009.

[58] H. Fu, M. K. Ng, M. Nikolova, and J. L. Barlow, “Efficient minimization methods of mixed ℓ_2-ℓ_1 and ℓ_1-ℓ_1 norms for image restoration,” *SIAM Journal on Scientific computing*, vol. 27, no. 6, pp. 1881–1902, 2006.

[59] D. Goldfarb and W. Yin, “Second-order cone programming methods for total variation-based image restoration,” *SIAM Journal on Scientific Computing*, vol. 27, no. 2, pp. 622–645, 2005.

[60] T. Goldstein and S. Osher, “The split Bregman method for ℓ_1 regularized problems,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[61] T. Goldstein, X. Bresson, and S. Osher, “Geometric applications of the split Bregman method: Segmentation and surface reconstruction,” *Journal of Scientific Computing*, vol. 45, no. 1-3, pp. 272–293, 2010.

[62] G. H. Golub and C. F. Van Loan, *Matrix Computations.* JHU Press, 2012, vol. 3.

[63] M. Grant and S. Boyd, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[64] ——, “CVX: Matlab software for disciplined convex programming, version 2.0 beta,” http://cvxr.com/cvx, Sep. 2012.

[65] R. Gribonval and M. Nielsen, “Sparse representations in unions of bases,” *Information Theory, IEEE Transactions on*, vol. 49, no. 12, pp. 3320–3325, 2003.

[66] M. R. Hestenes, “Multiplier and gradient methods,” *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[67] M. Hintermüller and G. Stadler, “An infeasible primal-dual algorithm for total bounded variation–based inf-convolution-type image restoration,” *SIAM Journal on Scientific Computing*, vol. 28, no. 1, pp. 1–23, 2006.

[68] Y. Hu and M. Jacob, “Higher degree total variation (HDTV) regularization for image recovery,” *Image Processing, IEEE Transactions on*, vol. 21, no. 5, pp. 2559–2571, 2012.

[69] B. Huang, S. Ma, and D. Goldfarb, “Accelerated linearized Bregman method,” *Arxiv preprint arXiv:1106.5413*, 2011.

[70] C. G. Koay, J. E. Sarlls, and E. Özarslan, “Three-dimensional analytical magnetic resonance imaging phantom in the Fourier domain,” *Magnetic Resonance in Medicine*, vol. 58, no. 2, pp. 430–436, 2007.

[71] K. Koh, S.-J. Kim, and S. P. Boyd, “An interior-point method for large-scale ℓ_1-regularized logistic regression.” *Journal of Machine learning research*, vol. 8, no. 8, pp. 1519–1555, 2007.

[72] R. Larson, *Elementary Linear Algebra, 7th Edition.* Brooks/Cole, Cengage Learning, 2013.

[73] Y. J. Lee, S. Lee, and J. Yoon, “A framework for moving least squares method with total variation minimizing regularization,” *Journal of Mathematical Imaging and Vision*, pp. 1–17, 2013.

[74] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[75] C. Louchet and L. Moisan, “Total variation as a local filter,” *SIAM Journal on Imaging Sciences*, vol. 4, no. 2, pp. 651–694, 2011.

[76] W.-S. Lu, *Digital Signal Processing III Lecture Notes.* ELEC 459/534 Course Notes, University of Victoria, 2011.

[77] ——, *Selected Topics in Digital Signal Processing: Sparse Signal Processing and Compressed Sensing.* ELEC 639A Course Notes, University of Victoria, 2011.

[78] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic resonance in medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[79] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 72–82, 2008.

[80] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way, Third Edition.* Elsevier Inc., 2009.

[81] G. O. Mohler, A. L. Bertozzi, T. A. Goldstein, and S. J. Osher, "Fast TV regularization for 2D maximum penalized likelihood estimation," *Journal of Computational and Graphical Statistics*, vol. 20, no. 2, pp. 479–491, 2011.

[82] J. L. Mueller and S. Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications.* SIAM, 2012, vol. 10.

[83] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[84] D. Needell, "Noisy signal recovery via iterative reweighted ℓ_1-minimization," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on.* IEEE, 2009, pp. 113–117.

[85] Y. Nesterov, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.

[86] ——, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," in *Soviet Mathematics Doklady*, 1983, pp. 372–376.

[87] M. K. Ng, R. H. Chan, and W.-C. Tang, "A fast algorithm for deblurring models with Neumann boundary conditions," *SIAM Journal on Scientific Computing*, vol. 21, no. 3, pp. 851–866, 1999.

[88] M. Nikolova, “Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers,” *SIAM Journal on Numerical Analysis*, vol. 40, no. 3, pp. 965–994, 2002.

[89] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, “An iterative regularization method for total variation-based image restoration,” *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005.

[90] S. Osher, Y. Mao, B. Dong, and W. Yin, “Fast linearized Bregman iteration for compressive sensing and sparse denoising,” *arXiv preprint arXiv:1104.0262*, 2011.

[91] J. K. Pant, “Compressive sensing using ℓ_p optimization,” Ph.D. dissertation, University of Victoria, 2012.

[92] J. K. Pant, A. Antoniou, and W.-S. Lu, “Reconstruction of sparse signals by minimizing re-weighted approximate l0-norm in the null space of the measurement matrix,” in *International Midwest Symposium on Circuits and Systems.* IEEE, 2010, pp. 430–433.

[93] G. Passty, “Ergodic convergence to a zero of the sum of monotone operators in Hilbert space,” *Journal of Mathematical Analysis and Applications*, vol. 72, no. 2, pp. 383–390, 1979.

[94] Z. Peng, M. Yan, and W. Yin, “Parallel and distributed sparse optimization,” in *Signals, Systems and Computers, 2013 Asilomar Conference on.* IEEE, 2013, pp. 659–646.

[95] Y. Rivenson and A. Stern, “Compressed imaging with a separable sensing operator,” *Signal Processing Letters, IEEE*, vol. 16, no. 6, pp. 449–452, 2009.

[96] R. T. Rockafellar, “A dual approach to solving nonlinear programming problems by unconstrained optimization,” *Mathematical Programming*, vol. 5, no. 1, pp. 354–373, 1973.

[97] R. Rockafellar, *Convex Analysis.* Princeton University Press, Princeton, NJ, 1970.

[98] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[99] Y. Saad, *Iterative Methods for Sparse Linear Systems.* SIAM, 2003.

[100] I. Selesnick and M. Figueiredo, "Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors," in *Proceedings of SPIE*, vol. 7446. Citeseer, 2009, p. 74460D.

[101] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," *Nuclear Science, IEEE Transactions on*, vol. 21, no. 3, pp. 21–43, 1974.

[102] S. L. Shishkin, H. Wang, and G. S. Hagen, "Total variation minimization with separable sensing operator," in *Image and Signal Processing.* Springer, 2010, pp. 86–93.

[103] X.-C. Tai and C. Wu, "Augmented lagrangian method, dual methods and split bregman iteration for rof model," in *Scale space and variational methods in computer vision.* Springer, 2009, pp. 502–513.

[104] H. L. Taylor, S. C. Banks, and J. F. McCoy, "Deconvolution with the ℓ_1 norm," *Geophysics*, vol. 44, no. 1, pp. 39–52, 1979.

[105] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[106] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, 2004.

[107] ——, "Just relax: Convex programming methods for identifying sparse signals in noise," *Information Theory, IEEE Transactions on*, vol. 52, no. 3, pp. 1030–1051, 2006.

[108] J. Tropp and S. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[109] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM Journal on Scientific Computing*, vol. 17, no. 1, pp. 227–238, 1996.

[110] Y. Wang, J. Yang, W. Yin, and Y. Zhang, “A new alternating minimization algorithm for total variation image reconstruction,” *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 248–272, 2008.

[111] B. Wohlberg and P. Rodriguez, “An iteratively reweighted norm algorithm for minimization of total variation functionals,” *Signal Processing Letters, IEEE*, vol. 14, no. 12, pp. 948–951, 2007.

[112] S. Wright, R. Nowak, and M. Figueiredo, “Sparse reconstruction by separable approximation,” *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2479–2493, 2009.

[113] Y. Xu, W. Yin, and S. Osher, “Learning circulant sensing kernels,” DTIC Document, Tech. Rep., 2012.

[114] J. Yan and W.-S. Lu, “Smoothed ℓ_p-ℓ_2 solvers for signal denoising,” in *ICASSP 2012*, Kyoto, Japan, Mar. 2012.

[115] ——, “Power-iterative strategy for ℓ_p-ℓ_2 optimization for compressive sensing: towards global solution,” in *Asilomar 2011*, Pacific Grove, CA, Nov. 2011.

[116] ——, “New algorithms for sparse representation of discrete signals based on ℓ_p-ℓ_2 optimization,” in *PacRim 2011*, pp. 73-78, Victoria, BC, Aug. 2011.

[117] ——, “Image denoising by generalized total variation regularization and least squares fidelity,” *Multidimensional Systems and Signal Processing*, 2013.

[118] ——, “Fast dual-based linearized Bregman algorithm for compressive sensing of digital images,” in *Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing (VIIP 2012)*, Banff, Canada, July 2012, pp. 43–49.

[119] W. Yin, “Analysis and generalizations of the linearized Bregman method,” *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 856–877, 2010.

[120] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, “Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing,” *SIAM J. Imaging Sci*, vol. 1, no. 1, pp. 143–168, 2008.

[121] W. Yin, S. Morgan, J. Yang, and Y. Zhang, "Practical compressive sensing with Toeplitz and circulant matrices," in *Proc. SPIE*, vol. 7744, 2010, p. 77440K.

[122] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *arXiv preprint arXiv:1310.7063*, 2013.

[123] S. Zhang and S. Mallat, "Matching pursuit with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[124] M. Zibulevsky and M. Elad, "ℓ_1-ℓ_2 optimization in signal and image processing," *Signal Processing Magazine, IEEE*, vol. 27, no. 3, pp. 76–88, 2010.