

RICE UNIVERSITY

**An Efficient Algorithm For Total Variation Regularization
with Applications to the Single Pixel Camera and
Compressive Sensing**

by

Chengbo Li

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Arts

APPROVED, THESIS COMMITTEE:

Yin Zhang, Professor, Chair
Computational and Applied Mathematics

William W. Symes, Noah G. Harding Professor
Computational and Applied Mathematics

Wotao Yin, Assistant Professor
Computational and Applied Mathematics

Kevin Kelly, Associate Professor
Electrical and Computer Engineering

HOUSTON, TEXAS

SEPTEMBER 2009

Abstract

An Efficient Algorithm For Total Variation Regularization with Applications to the Single Pixel Camera and Compressive Sensing

by

Chengbo Li

In this thesis, I propose and study an efficient algorithm for solving a class of compressive sensing problems with total variation regularization. This research is motivated by the need for efficient solvers capable of restoring images to a high quality captured by the single pixel camera developed in the ECE department of Rice University. Based on the ideas of the augmented Lagrangian method and alternating minimization to solve subproblems, I develop an efficient and robust algorithm called TVAL3. TVAL3 is compared favorably with other widely used algorithms in terms of reconstruction speed and quality. Convincing numerical results are presented to show that TVAL3 is suitable for the single pixel camera as well as many other applications.

Acknowledgements

I would like to express my deep and sincere gratitude to my academic advisor, Professor Yin Zhang. His enthusiasm, profound knowledge, and upbeat personality have greatly influenced me. He has been helping me accumulate my research skills, tap into my full potential, as well as build up my confidence step by step in conducting research. Without his wholehearted guidance, I might have already lost my interest in optimization. I really take pride in working with him.

I would also like to extend my deepest appreciation to Dr. Wotao Yin, who led me to the united CAAM family at Rice University around two years ago. He has provided me tremendous help on both academic and living aspects. I owe many thanks to him for his encouragement, patience, and guidance. Besides, his intelligence and humor have deeply impressed me. He is not only my mentor, but also one of my best friends.

Ting Sun, who is one of my collaborators in the ECE department of Rice University, has shared large quantities of data with me and helped me fully understand the single pixel camera and other related electrical engineering knowledge. I would like to say that collaborating with Ting Sun has effectively accelerated my research. I owe her many thanks.

I feel so grateful for my dear friend Josh Bell and his family. They have helped me revise the draft of my thesis word by word—picking out all catachreses and solecisms. It is extremely a boring and painful process for those who know nothing about professional knowledge. What they have done for me will always be remembered. I also appreciate all my friends who are supportive and dedicated all the time.

I need to thank Dr. Tapia who taught me that mathematicians could take on more than mathematics; Dr. Borcea, who was my mentor during my first year at CAAM

and helped me adapt the new environment; Dr. Hewitt who offered suggestions of my usage of the English Language and recommendations for my presentations on several occasions; my committee members Dr. Symes and Dr. Kelly who earnestly reviewed my thesis. Besides, I feel grateful for all other professors who have provided me knowledge and expertise during my undergraduate and graduate studies.

Last but certainly not least, I am grateful for my parents, grandfather, and grandmother for their selfless love and unconditional support throughout these years. You are the love of my entire life.

I am willing to dedicate this thesis to my dear Yun to commemorate all things between us.

Contents

Abstract	ii
Acknowledgements	iii
List of Figures	vii
1 Introduction	1
1.1 Compressive Sensing Background	2
1.1.1 Greedy Algorithms	2
1.1.2 ℓ_1 Minimization	3
1.1.3 TV Minimization	4
1.2 Single Pixel Camera	6
1.3 Methodologies of TV Solvers	8
2 TVAL3 Scheme and Algorithms	11
2.1 Augmented Lagrangian Method Review	12
2.2 Augmented Lagrangian Algorithm for TV Minimization	18
2.3 Alternating Direction Algorithm for the Subproblem	20
2.3.1 Shrinkage-like Formulas	21
2.3.2 One-step Steepest Descent Scheme	26
2.4 Overall Algorithm and Extensions	29
3 Fast Walsh Hadamard Transform	32
3.1 Hadamard Matrix	33
3.2 Kronecker Product and Fast Walsh Hadamard Transform	37
3.3 Comparisons	41
4 Numerical Results and Discussions	44
4.1 State-of-the-art Solvers and Test Platform	44
4.2 Comparisons Based on Synthetic Data	46
4.3 Comparisons Based on Measured Data	55
4.4 Initial Tests on Complex Signals and Nonnegativity Constraints	61
4.5 Discussions	63

5	Future Work	65
5.1	Hyperspectral Imaging	66
5.1.1	Basic Concepts	66
5.1.2	Initial Formulation	68
5.1.3	Parallel Algorithms and Implementations on High Performance Computers	71
5.2	Exploration on Dual Method	72
5.2.1	Derivation of Dual Problem	73
5.2.2	Methodology on Dual Problem	76
	Bibliography	78

List of Figures

1.1	Single pixel camera block diagram	6
3.1	Running time comparison between two FWHT implementations . . .	42
3.2	Running time comparison between FWHT and FFT	43
4.1	Reconstructed 1D staircase signal	46
4.2	Recoverability for 1D staircase signals	47
4.3	Recovered phantom image from orthonormal measurements	49
4.4	Recovered phantom image from non-orthonormal measurements . . .	50
4.5	Recovered MR brain image	52
4.6	Recoverability for MR brain image	53
4.7	Real target in visible light	56
4.8	Recovered infrared RI image	58
4.9	Recovered the transistor image	59
4.10	Recovered 1D complex staircase signal	62
4.11	Recovered CT thorax image	62

Chapter 1

Introduction

This thesis concentrates on developing an efficient algorithm which solves a well-known compressive sensing (also known as compressed sensing or CS) problem with total variation (TV) regularization. The main application of this algorithm is to reconstruct the high-resolution image captured by a single pixel camera (SPC). The basic questions are: what is the background and motivation of this research, what methods are used, why is a new algorithm necessary, and how does this new algorithm behave compared with other existing solvers or algorithms? All of these questions will be answered step by step in this thesis.

The basic background including compressive sensing and single pixel camera, existing reconstruction algorithms, and the general methodology are introduced in this chapter. The second chapter, one of the most essential chapters in this thesis, describes the main algorithm in detail and introduces the corresponding solver TVAL3 [98]. A structured measurement matrix correlating to the single pixel camera and how this measurement matrix is able to improve the algorithm will be discussed in the following chapter. The algorithm described in this thesis compares favorably with several state-of-the-art algorithms in the fourth chapter of this thesis. Numerical re-

sults and the following discussion will also be covered. Last but not least, some related topics such as the TV minimization algorithm for dual problems and hyperspectral imagery which will require further research during my Ph.D. studies, are proposed in the last chapter.

1.1 Compressive Sensing Background

Compressive sensing [4] is a technique which reconstructs or obtains a sparse or compressible signal. A large but sparse signal is encoded by a relatively small number of linear measurements, and then the original signal is recovered from the encoded one. It has been proven that computing the sparsest solution directly generally requires prohibitive computations of exponential complexity [46], so several heuristic methods have been developed, such as Matching Pursuit [51], Basis Pursuit [53, 54], log-barrier method [55], iterative thresholding method [57, 58], and so forth. Most of these methods or algorithms fall into three distinct categories: greedy algorithms, ℓ_1 minimization, and TV minimization.

1.1.1 Greedy Algorithms

Generally speaking, a greedy algorithm refers to any algorithm following the meta-heuristic of choosing the best immediate or local optimum at each stage and expecting to find the global optimum at the end. It can find the global optimum for some optimization problems, but not for all [50]. Mallat and Zhang [51] introduced Matching Pursuit (MP) in 1993, which is the prototypical greedy algorithm applied to compressive sensing. This algorithm decomposes any signal into a linear combination of waveforms in a redundant dictionary of functions so that selected waveforms optimally match the structure of the signal. MP is easy to implement and has an exponential

rate of convergence [66] and good approximation properties [65]. However, there is no theoretical guarantee that MP can achieve sparse representations. Pati *et al.* propose a variant of MP, Orthogonal Matching Pursuit (OMP) [52], which guarantees the nearly sparse solution under some conditions [67]. A primary drawback of MP and its variants is the incapability of attaining truly sparse representations. The failure is usually caused by an inappropriate initial guess. This shortcoming also motivated the development of algorithms based on ℓ_1 minimization.

1.1.2 ℓ_1 Minimization

In 1986, Santosa and Symes [7] suggested ℓ_1 minimization to recover sparse spike trains for the first time. In the next few years, Donoho and his colleague [8, 9] also discovered some early results related to ℓ_1 minimization for signal recovery. The question why ℓ_1 minimization could work in some special setups was further investigated and answered in a series of paper [10, 11, 12, 13, 14, 15].

Grounded on those early efforts, a new CS theory was proposed by Candès, Tomberg, Tao [2, 3], and Donoho [4] in 2006, which theoretically guarantees ℓ_1 minimization is equivalent to ℓ_0 minimization under some conditions on signal reconstruction. Specifically, they claim that a signal which is K -sparse under some basis can be exactly recovered from cK linear measurements by ℓ_1 minimization under some conditions, where c is a constant. The new CS theory has significantly improved those earlier results. How big the constant c is here directly decides the size of linear measurements, important information needed to encode or decode a signal. The introduction of the concept *restricted isometry property* (RIP) for matrices [1, 4] gives the theoretical response. E. Candès, Tao, and Donoho prove that if the measurements satisfy the RIP of a certain degree, it is sufficient to recover the sparse signal exactly

from its decoded signal. However, it is extremely difficult to verify the RIP property in practice. Fortunately, Candès *et al.* show that RIP holds with high probability when the measurements are random. However, is RIP truly an indispensable property for CS analysis? For instance, measurement matrices A and GA in ℓ_1 minimization should result in exactly the same recoverability and stability as long as matrix G is square and nonsingular, but their RIP could vary a lot. A non-RIP analysis, studied by Y. Zhang [5], proves recoverability and stability theorems without the aid of RIP and clarifies prior knowledge can never hurt but possibly enhance recovery via ℓ_1 minimization. Usually ℓ_1 minimization algorithms require fewer measurements than greedy algorithms. Basis Pursuit (BP) [53, 54], which seeks the solution that minimizes the ℓ_1 norm of the coefficients, is a prototype of ℓ_1 minimization. BP can simply be comprehended as linear programming solved by some standard methods. Furthermore, BP can compute sparse solutions in situations where greedy algorithms fail [54].

All this work enriches the significance of studying and applying ℓ_1 minimization and compressive sensing in practice. The related studies [21, 22, 23, 27, 28] have also inspired the flourishing research in the compressive sensing area. Many applications have been studied, such as reconstruction or denoising of Magnetic Resonance Images (MRI) [29, 30], analog-to-information conversion [31], sensor networks [34, 35], and even homeland security [68].

1.1.3 TV Minimization

In the broad area of compressive sensing, ℓ_1 minimization has attracted intensive research activities since the discovery of ℓ_0/ℓ_1 equivalence. However, for image restoration, recent research has confirmed that the use of total variation (TV) regularization

instead of the ℓ_1 term in CS problems makes the recovered image quality sharper by preserving the edges or boundaries more accurately, which is essential to characterize images. The advantages of TV minimization stem from the property that it can recover not only sparse signals or images, but also dense staircase signals or piecewise constant images. In other words, TV regularization would succeed when the gradient of the underlying signal or image is sparse. Even though this result has only been theoretically proven under some special circumstances [3], it stands true on a much larger scale empirically.

Rudin, Osher, and Fatemi [6] first introduced the concept total variation for image denoising in 1992. From then on, total variation minimizing models have become one of the most popular and successful methodologies for image restoration. A detailed discussion on TV models has been reported by Chambolle *et al.* [25, 26]. However, the properties of non-differentiability and non-linearity of TV functions make them far less accessible computationally than solving ℓ_1 minimization models. Geman and Yang [33] proposed a joint minimization method to solve half-quadratic models [32, 33], which are variants of TV models. Grounded on half-quadratic models, Wang, Yang, Yin, and Zhang applied TV minimization to deconvolution and denoising problems [18] and successfully extended their idea to image reconstruction [36] and multichannel image deblurring or denoising problems [37, 38]. Their reconstruction algorithm for TV minimization is very efficient and effective, but it restricts the measurement matrix to the partial Fourier matrix. In 2004, Chambolle [24] proposed an iterative algorithm for TV denoising and proved the linear convergence. Furthermore, Chambolle's algorithm can be extended to solve image reconstruction problems with TV regularization while the measurement matrix is orthogonal.

Due to the powerful application of TV regularization in the edge-detection and many other fields, researchers kept trying for several years to explore algorithms for

solving TV minimization problems. However, these algorithms are still either much slower or less robust compared with algorithms designed for ℓ_1 minimization. The algorithm proposed in this thesis has successfully overcome this difficulty and led to a new solver (named TVAL3) for TV minimization which is as fast as or even faster than most ℓ_1 minimization algorithms and accepts a vast range of measurement matrices.

1.2 Single Pixel Camera

A significant application of compressive sensing in recent years is the successful design of the single pixel camera. This concept was initially proposed by Baraniuk, Kelly, *et al.* [39]. As shown in Figure 1.1, this new-concept camera is mainly composed of two

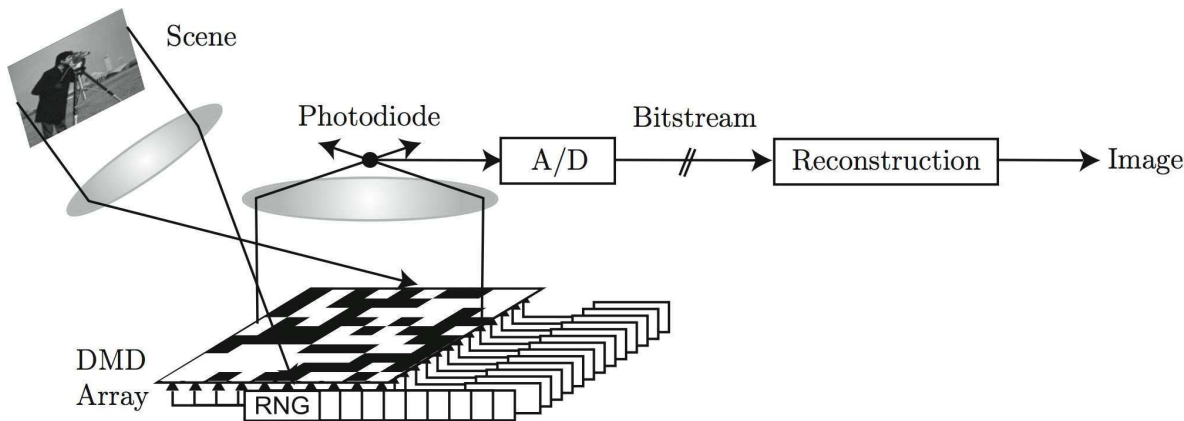


Figure 1.1: Single pixel camera block diagram [39].

devices: the *digital micro-mirror device* (DMD) [43] and the *photodiode* (PD). The desired image (camera man) is projected on a DMD array which is fabricated by $m \times n$ little mirrors and oriented in the pseudorandom pattern decided by *random number generators* (RNG). Then the lightfield goes through a lens and converges to a single PD by which one pixel value is obtained. Each different mirror pattern produces one measurement. Repeating this process M times, M pixel values corresponding

to M measurements are captured. A sparse approximation to the original image can be recovered from known pixel values and random measurements by means of compressive sensing techniques. Some extended research related to a single pixel camera has been done including infrared imaging [44], laser-based failure-analysis [45], and others [40, 41, 42] at Rice University.

Why should people care about the single pixel camera considering the fact that the traditional digital camera with ten mega pixels is ubiquitous and low-priced? As a matter of fact, imaging at wavelengths where silicon is blind is much more complicated and costly than imaging at visual wavelengths. This results in the unaffordable price of a digital camera for infrared with comparable resolution. On the other hand, the infrared camera has wide applications in industrial, military, and medical domains, such as heat energy detection, night vision, internal organ examination, and so on. The manufacture of single pixel infrared cameras could greatly decrease in price so as to be affordable for everyone and applicable everywhere. All of these reasons motivate researchers to focus on the development of the single pixel camera with respect to both hardware and software. Here, the software refers to the core recovery solver. An efficient and robust solver, which is able to reconstruct a clean and sharp image in a relatively short time, is intensely expected.

Because the number of measurements M is much less than the original resolution while dealing with the desired image using the single pixel camera, it is natural to model the recovery process as a compressive sensing problem. Thus, compressive sensing algorithms can be applied to the single pixel camera. Before the emergence of TVAL3, which is the new solver based on the algorithm described in this thesis, the single pixel camera adopted ℓ_1 -Magic [3, 2, 1] and FPC [17] as the core recovery solver. Solvers for ℓ_1 minimization and TV minimization are named ℓ_1 solvers and TV solvers respectively. ℓ_1 -Magic, implemented by Candès and Romberg, is one of pioneer TV

solvers for compressive sensing. It was the initial solver to recover images for the single pixel camera due to its good reputation for stability and edge-preservation. However, the disadvantage is the much longer reconstructing time compared with ℓ_1 solvers. For instance, it is impractical to deal with an image whose resolution is 512×512 using ℓ_1 -Magic. In contrast, as one of the fastest ℓ_1 solvers, FPC [17] implemented by Hale, Yin, and Zhang is capable of recovering the high-resolution image in a relatively short time. However, as mentioned before, the edges of images recovered ℓ_1 solvers cannot be preserved as well as those recovered by TV solvers, especially when high noise level exists. Besides, wavelet transformation is necessary for ℓ_1 solvers, but not for TV solvers. Thus, the single pixel camera highly desires a high-quality TV solver whose running time is comparable with ℓ_1 solvers.

1.3 Methodologies of TV Solvers

Contrary to abundant ℓ_1 solvers, only a limited number of TV solvers are available. To the best of my knowledge, only SOCP [19], ℓ_1 -Magic [3, 2, 1], TwIST [57, 58], NESTA [56], and RecPF [36] are publicly available for image reconstruction with TV regularization.

The approach behind SOCP solver is to reformulate TV minimization as a second-order cone program, which is solvable by interior-point algorithms. This solver is easy to adapt various convex TV models with distinct terms and constraints and able to achieve high accuracy. However, it is very slow since SOCP embeds the interior-point algorithm and directly solves a linear system at each iteration.

Similar to SOCP, ℓ_1 -Magic also focuses on second-order cone reformulation of TV models, but it is implemented by the log-barrier method. At each log-barrier iteration, Newton's method proceeds with the approximate solution at the last iteration as the

initial guess. Compared with SOCP, ℓ_1 -Magic solves the linear system in an iterative way, which is more efficient than directly solving the linear system. However, applying Newton's method at each iteration is still time-consuming when facing a large-scale problem.

In the last few years, iterative shrinkage/thresholding (IST) algorithms were independently proposed by several authors [60, 61, 62, 63, 64]. IST is able to minimize CS models with some non-quadratic and non-smooth regularization terms. The convergence rate of IST algorithms highly relies on the linear observation operator. TwIST implements a nonlinear second-order iterative version of IST algorithms, which exhibits much faster convergence rate than IST when the linear observation operator is ill-conditioned. This solver can also be regarded as alternating algorithm of two steps, one of which is a denoising step. For TV minimization, Chambolle's denoising algorithm [24] is coupled to TwIST. Chambolle's algorithm is an iterative fixed point algorithm based on a dual formulation. This scheme converges quite fast at the first iteration, sometimes bringing on a visually satisfactory result, but the remaining iterations tend to be quite a slow convergence. The denoising step is the dominating time-consuming part while running TwIST. Therefore, the efficiency of Chambolle's algorithm mostly determines the efficiency of TwIST.

In April 2009, Bobin, Becker, and Candès developed a new solver NESTA, a first-order method of solving BP problems. They were notably inspired by Nesterov's smoothing technique [16], whose essential idea is a subtle averaging of sequences of iterates. Their algorithm is easily extended to TV minimization by slightly modifying the smooth approximation of the objective function. However, the current version of NESTA still requires that $A^T A$ is an orthogonal projector where A represents the measurement matrix. Further investigation may extend this method to the non-orthogonal cases as indicated in their paper [56].

As mentioned before, Wang, Yang, Yin, and Zhang [18] have proposed a new alternating minimization method for deconvolution and denoising problems with TV regularization. The key feature of this algorithm is the splitting idea, which is brought to approximate the TV regularization. Yang, Zhang, and Yin [36] extended the same scheme to the compressive sensing area and implemented the solver RecPF. A distinct merit of this solver is low cost at each iteration, which requires only two matrix-vector multiplications per iteration as the dominant computation. As a TV solver, RecPF is competitive in speed to most ℓ_1 solvers, which is a surprising discovery motivating my work on the new TV algorithm, but it can only accept the partial Fourier matrix as its measurements.

The splitting idea originated from [18] is also the springboard to exploit a new efficient and robust TV solver which is able to lead the single pixel camera one step closer to practical application. A detailed description of the algorithm will be given in next chapter.

Chapter 2

TVAL3 Scheme and Algorithms

A chief contribution of this thesis is regarded as proposing a new efficient TV minimization scheme based on augmented Lagrangian and alternating direction algorithms, short for “TVAL3 scheme”. It is presented in detail in this chapter for solving the compressive sensing problem with total variation regularization:

$$\min_u \sum_i \|D_i u\|, \quad \text{s.t. } Au = b, \quad (2.1)$$

where $u \in \mathbb{R}^n$ or $u \in \mathbb{R}^{s \times t}$ with $s \cdot t = n$, $D_i u \in \mathbb{R}^2$ is the discrete gradient of u at pixel i , $A \in \mathbb{R}^{m \times n}$ ($m < n$) is the measurement matrix, and $f \in \mathbb{R}^m$ is the observation of u via some linear measurements. $\|\cdot\|$ can be either 1-norm (corresponding to the anisotropic TV) or 2-norm (corresponding to the isotropic TV). TVAL3 scheme is able to handle different boundary conditions for u , such as periodic, Neumann, and other boundary conditions. The periodic boundary condition is used here to calculate $\sum_i \|D_i u\|$ for simplicity.

This model (2.1) is very difficult to solve directly due to the non-differentiability and non-linearity of the TV term. The algorithm proposed in this chapter is derived

from the classic approach of alternating direction method [69], or ADM, that minimizes augmented Lagrangian functions [70, 71] through an alternating minimization scheme and updates multipliers after each sweep. The convergence of such algorithms has been well analyzed in the literature (see [81], for example, and the references therein).

The background of the augmented Lagrangian method is reviewed in Section 2.1 and the TVAL3 scheme is developed step by step in Section 2.2, 2.3, and 2.4.

2.1 Augmented Lagrangian Method Review

For constrained optimization, an influential class of methods seeks the minimizer or maximizer by approaching the original constrained problem by a sequence of unconstrained subproblems. The quadratic penalty method which could be regarded as the precursor to the augmented Lagrangian method, should be traced back to Courant [20] in 1943. This method puts a quadratic penalty term instead of the constraint in the objective function where each penalty term is a square of the constraint violation with the multiplier. Due to its simplicity and intuitive appeal, this approach is widely used. However, it requires multipliers to go to infinity to guarantee the convergence, which may cause the ill-conditioning problem numerically. In 1969, Hestenes [70] and Powell [71] independently proposed the augmented Lagrangian method which successfully avoided this inherent problem by introducing explicit Lagrangian multiplier estimates at each iteration into the objective function.

Let us begin with considering the equality-constrained problem

$$\min_x f(x), \quad \text{s.t. } h(x) = 0, \quad (2.2)$$

where h is a vector-valued function and both f and h_i for all i are differentiable. The first-order optimality conditions are

$$\nabla \mathcal{L}(x, \lambda) = 0, \quad (2.3)$$

$$h(x) = 0, \quad (2.4)$$

where $\mathcal{L}(x, \lambda) = f(x) - \lambda^T h(x)$. We say the *linear independence constraint qualification* (LICQ) holds at the point x^* if and only if the set $\{\nabla h_i(x^*)\}$ is linearly independent. The optimality conditions are necessary for the optimal points of (2.2) if LICQ holds there. When the primal problem (2.2) is convex, the optimality conditions become also sufficient.

In light of the optimality conditions, a solution x^* to the primal problem (2.2) is both a stationary point of the Lagrangian function and a feasible point of the constraint, which means x^* solves

$$\min_x \mathcal{L}(x, \lambda), \quad \text{s.t. } h(x) = 0. \quad (2.5)$$

According to the idea of the quadratic penalty method, it is likely to make x^* an unconstrained minimizer by penalizing the constraint violations. For example, it may approximately solve

$$\min_x \mathcal{L}_A(x, \lambda; \mu) = f(x) - \lambda^T h(x) + \frac{\mu}{2} h(x)^T h(x).$$

Minimizing this alternate problem is well-known as an augmented Lagrangian method, and $\mathcal{L}_A(x, \lambda; \mu)$ is called the augmented Lagrangian function.

The augmented Lagrangian function differs from the standard Lagrangian function by adding a square penalty term, and differs from the quadratic penalty function

by the presence of the linear term involving the multiplier λ . In this respect, the augmented Lagrangian function is a combination of the Lagrangian and quadratic penalty functions.

An iterative algorithm implementing the augmented Lagrangian method will be described next. Fixing the multiplier λ at the current estimate λ^k and the barrier parameter μ to $\mu^k > 0$ at the k th iteration, we minimize the augmented Lagrangian function $\mathcal{L}_A(x, \lambda^k; \mu^k)$ with respect to x and denote the minimizer as x^{k+1} . Hestenes [70] and Powell [71] have suggested formula

$$\lambda^{k+1} = \lambda^k - \mu^k h(x^{k+1}), \quad (2.6)$$

in order to update the multiplier estimates from iteration to iteration and they have proven the convergence of the generated sequence to the true multiplier λ^* .

This discussion motivates the following algorithmic framework [78]:

Algorithm 1 (Augmented Lagrangian Method).

Initialize μ^0, λ^0 , tolerance tol , and starting point x^0 ;

While $\|\nabla \mathcal{L}(x^k, \lambda^k)\| > tol$ **Do**

Set $x_0^{k+1} = x^k$;

Find minimizer x^{k+1} of $\mathcal{L}_A(x, \lambda^k; \mu^k)$, starting from x_0^{k+1}

and terminating when $\|\nabla_x \mathcal{L}_A(x, \lambda^k; \mu^k)\| \leq tol$;

Update the multiplier using (2.6) *to obtain* λ^{k+1} ;

Choose the new penalty parameter $\mu^{k+1} \geq \mu^k$;

End Do

At each iteration, we theoretically achieve

$$\nabla_x \mathcal{L}_A(x^{k+1}, \lambda^k; \mu^k) = 0.$$

This can be expanded as

$$\nabla f(x^{k+1}) - \nabla h(x^{k+1})\lambda^k + \mu^k \nabla h(x^{k+1})h(x^{k+1}) = 0,$$

which is equivalent to

$$\nabla f(x^{k+1}) - \nabla h(x^{k+1})[\lambda^k - \mu^k h(x^{k+1})] = 0.$$

Following the update formula of multiplier estimates (2.6), this can be rearranged as

$$\nabla f(x^{k+1}) - \nabla h(x^{k+1})\lambda^{k+1} = 0,$$

which is the variant of

$$\nabla \mathcal{L}(x^{k+1}, \lambda^{k+1}) = 0.$$

This equation means the optimality conditions for (2.5) are partially satisfied. Therefore, Algorithm 1 terminates while

$$\nabla_{\lambda} \mathcal{L}(x^{k+1}, \lambda^{k+1}) = -h(x^{k+1}) = 0,$$

or in practice,

$$\|h(x^{k+1})\| \leq tol.$$

Some basic properties of the augmented Lagrangian method will be reviewed next. The following result given by Bertsekas [79, 80] provides a precise mathematical description on some error bounds which help quantify the rate of convergence.

Theorem 1 (Local Convergence Theorem). *Let x^* be a local solution of (2.2) at which the gradients $\nabla h_i(x^*)$ are linearly independent, and the second-order sufficient*

conditions are satisfied for $\lambda = \lambda^*$; i.e., $\nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*)$ is positive definite. Choose $\bar{\mu} > 0$ so that $\nabla_{xx}^2 \mathcal{L}_A(x^*, \lambda^*; \bar{\mu})$ is also positive definite. Then there exist positive constants δ , ϵ , and M such that the following claims hold:

1. For all $(\lambda^k, \mu^k) \in \mathcal{D}$ where $\mathcal{D} \triangleq \{(\lambda, \mu) : \|\lambda - \lambda^*\| < \delta\mu, \mu \geq \bar{\mu}\}$, the problem

$$\min_x \mathcal{L}_A(x, \lambda^k; \mu^k) \quad \text{s.t.} \quad \|x - x^*\| = \epsilon$$

has a unique solution x^k . It satisfies

$$\|x^k - x^*\| \leq \frac{M}{\mu^k} \|\lambda^k - \lambda^*\|.$$

Moreover, the function $x(\lambda, \mu)$ is continuously differentiable in the interior of \mathcal{D} .

2. For all $(\lambda^k, \mu^k) \in \mathcal{D}$,

$$\|\lambda^{k+1} - \lambda^*\| \leq \frac{M}{\mu^k} \|\lambda^k - \lambda^*\|,$$

where λ^{k+1} is attained by (2.6).

3. For all $(\lambda^k, \mu^k) \in \mathcal{D}$, $\nabla_{xx}^2 \mathcal{L}_A(x^k, \lambda^k; \mu^k)$ is positive definite and $\nabla h_i(x^k)$ are linearly independent.

A detailed proof for local convergence theorem can be found in [79], pp. 108.

The local convergence theorem implies three features of Algorithm 1. First, the algorithm converges in one iteration if $\lambda = \lambda^*$. Second, if μ^k is large enough to satisfy $\frac{M}{\mu^k} < 1$, the error bounds in the theorem are able to guarantee that

$$\|\lambda^{k+1} - \lambda^*\| < \|\lambda^k - \lambda^*\|;$$

i.e., the multiplier estimates converge linearly. Hence, $\{x^k\}$ also converges linearly. Last but not least, if $\lim \mu^k = +\infty$, then

$$\lim_{k \rightarrow +\infty} \frac{\|\lambda^{k+1} - \lambda^*\|}{\|\lambda^k - \lambda^*\|} = 0;$$

i.e., the multiplier estimates converge superlinearly.

The convergence rate mentioned above is not comparable to the other methods in general, because the augmented Lagrangian method requires solving an unconstrained minimization subproblem at each iteration, which is probably more expensive than the iterations of other methods. Thus, designing an elaborate scheme to solve the subproblem efficiently is one of the key issues while applying the augmented Lagrangian method.

In practice, it is unlikely to exactly solve the unconstrained minimization subproblem at each iteration. Rockafellar [72] has proven the global convergence in the convex case for an arbitrary penalty factor and without the requirement of an exact minimum at each iteration of the augmented Lagrangian method.

Theorem 2 (Global Convergence Theorem). *Suppose that*

1. (2.2) is a convex optimization problem; i.e., f is convex and h_i are linear constraints;
2. the feasible set $\{x : h(x) = 0\}$ is non-empty;
3. $\mu^k = \mu$ is constant for all k ;
4. a sequence $\{\epsilon_k\}_1^\infty$ satisfies $0 \leq \epsilon_k \rightarrow 0$ and

$$\sum_i^\infty \sqrt{\epsilon_k} < \infty.$$

Set tolerance to ϵ_k and update multiplier following (2.6) at iteration k in Algorithm 1. Then attained sequence $\{x^k\}$ converges to the global minimizer of (2.2).

A detailed proof for global convergence theorem can be found in [72], pp. 560–561.

This theorem confirms the global convergence in the convex case even though only approximate solutions for unconstrained subproblems are available in numerical computation and completes the theory of the augmented Lagrangian method.

Other than (2.6) proposed by Hestenes and Powell, Buys [73] and Tapia [74, 75] have suggested another two multiplier update formulas (called Buys update and Tapia update respectively) which both involve second-order information of $\mathcal{L}_A(x, \lambda; \mu)$. Tapia [76] and Byrd [77] have shown that both update formulas give quadratic convergence if one-step (for Tapia update) or two-step (for Buys update) Newton’s method is applied to minimizing the augmented Lagrangian function instead of the usual infinite number of steps for exact minimization. However, each step of Newton’s method can be computationally too expensive for applications in this thesis since it requires computing the Hessian of the augmented Lagrangian function.

2.2 Augmented Lagrangian Algorithm for TV Minimization

In stead of employing the augmented Lagrangian method to minimize the TV model (2.1) directly, we consider an equivalent variant of (2.1)

$$\min_{w_i, u} \sum_i \|w_i\|, \quad \text{s.t. } Au = b \text{ and } D_i u = w_i \text{ for all } i. \quad (2.7)$$

Its corresponding augmented Lagrangian function is

$$\begin{aligned} \mathcal{L}_A(w_i, u) = & \sum_i (\|w_i\| - \nu_i^T(D_i u - w_i) + \frac{\beta_i}{2} \|D_i u - w_i\|_2^2) \\ & - \lambda^T(Au - b) + \frac{\mu}{2} \|Au - b\|_2^2. \end{aligned} \quad (2.8)$$

Since (2.7) is still a convex problem, the global convergence theorem is able to guarantee the convergence while applying the augmented Lagrangian method to it. According to Algorithm 1 described above, ν_i and λ should be updated as long as (2.8) is minimized at each iteration. Let u^* and w_i^* represent the true minimizers of (2.8). In the light of (2.6), the update formulas of multipliers follow

$$\tilde{\nu}_i = \nu_i - \beta_i(D_i u^* - w_i^*) \quad \text{for all } i, \quad (2.9)$$

$$\tilde{\lambda} = \lambda - \mu(Au^* - b). \quad (2.10)$$

An alternating minimization algorithm for the image deconvolution and denoising has been proposed by Wang, Yang, Yin, and Zhang [18]. They introduced the variable-splitting technique to the compressive sensing area for the first time. In that paper, the TV regularization term is split into two terms with the aid of a new slack variable so that an alternating minimization scheme can be coupled to minimize the approximate objective function. The algorithm described in this thesis can also be derived under the variable-splitting technique.

If the augmented Lagrangian method is applied directly to (2.1), the corresponding augmented Lagrangian function is

$$\tilde{\mathcal{L}}_A(u) = \sum_i \|D_i u\| - \lambda^T(Au - b) + \frac{\mu}{2} \|Au - b\|_2^2. \quad (2.11)$$

If we introduce a slack variable $w_i \in \mathcal{R}^2$ at each pixel to transfer $D_i u$ out of the non-differentiable term $\|\cdot\|$ and penalize the difference between them, then it results in splitting every term in the first sum of (2.11) into three terms:

$$\|w_i\| - \nu_i^T(D_i u - w_i) + \frac{\beta_i}{2}\|D_i u - w_i\|_2^2.$$

Bringing these three terms back to (2.11) leads to the same objective function for the subproblem as (2.8).

The algorithmic framework of the augmented Lagrangian method indicates that it is essential to minimize $\mathcal{L}_A(w_i, u)$ efficiently at each iteration to solve (2.1). The subproblem is still hard to solve efficiently in a direct way due to the non-differentiability and non-linearity. Therefore, an iterative way is proposed in the next section—the alternation minimization scheme.

2.3 Alternating Direction Algorithm for the Subproblem

The subproblem is to minimize the augmented Lagrangian function; i.e.,

$$\begin{aligned} \min_{w_i, u} \mathcal{L}_A(w_i, u) &= \sum_i (\|w_i\| - \nu_i^T(D_i u - w_i) + \frac{\beta_i}{2}\|D_i u - w_i\|_2^2) \\ &\quad - \lambda^T(Au - b) + \frac{\mu}{2}\|Au - b\|_2^2. \end{aligned} \tag{2.12}$$

The alternating direction method [69], which was originally proposed to deal with parabolic and elliptic differential equations, is embedded here to solve (2.12) efficiently.

2.3.1 Shrinkage-like Formulas

Suppose that u_k and $w_{i,k}$ respectively denote the approximate minimizers of (2.8) at the k th iteration which refers to the inner iteration while solving the subproblem. Assuming that u_j and $w_{i,j}$ are available for all $j = 0, 1, \dots, k$, $w_{i,k+1}$ can be attained by

$$\begin{aligned} \min_{w_i} \mathcal{L}_A(w_i, u_k) &= \sum_i (\|w_i\| - \nu_i^T (D_i u_k - w_i) + \frac{\beta_i}{2} \|D_i u_k - w_i\|_2^2) \\ &\quad - \lambda^T (A u_k - b) + \frac{\mu}{2} \|A u_k - b\|_2^2, \end{aligned}$$

which is equivalent to solve the so-called “ w -subproblem”

$$\min_{w_i} \sum_i (\|w_i\| - \nu_i^T (D_i u_k - w_i) + \frac{\beta_i}{2} \|D_i u_k - w_i\|_2^2). \quad (2.13)$$

The w -subproblem is separable with respect to \mathbf{w}_i . In what follows, we argue that every separated problem admits a closed form solution.

Lemma 1. *For $x \in \mathbb{R}^p$, the subdifferential of $f(x) \triangleq \|x\|_1$ is given component by component*

$$(\partial f(x))_i = \begin{cases} \text{sgn}(x_i), & \text{if } x_i \neq 0; \\ \{h : |h| \leq 1, h \in \mathbb{R}\}, & \text{otherwise.} \end{cases}$$

The proof of Lemma 1 is easily extended from the subdifferential of absolute value in \mathbb{R} . Detailed proof is omitted here.

Lemma 2. *For given $\beta > 0$ and $\nu, y \in \mathbb{R}^q$, the minimizer of*

$$\min_x \|x\|_1 - \nu^T (y - x) + \frac{\beta}{2} \|y - x\|_2^2 \quad (2.14)$$

is given by the 1D shrinkage-like formula

$$x^* = \max \left\{ \left| y - \frac{\nu}{\beta} \right| - \frac{1}{\beta}, 0 \right\} \operatorname{sgn} \left(y - \frac{\nu}{\beta} \right). \quad (2.15)$$

Proof. Since the objective function is convex, bounded below and coercive, there exists at least one minimizer x^* for (2.14). According to the optimality condition for convex optimization, the origin should be included in the subdifferential of the objective function at the minimizer. In light of Lemma 1, each component x_i^* must satisfy

$$\begin{cases} \operatorname{sgn}(x_i) + \beta(x_i^* - y_i) + \nu_i = 0 & \text{if } x_i^* \neq 0; \\ |\nu_i - \beta y_i| \leq 1 & \text{otherwise.} \end{cases} \quad (2.16)$$

If $x_i^* \neq 0$, (2.16) gives us

$$x_i^* + \frac{\operatorname{sgn}(x_i)}{\beta} = y_i - \frac{\nu_i}{\beta},$$

which leads to

$$|x_i^*| + \frac{1}{\beta} = \left| y_i - \frac{\nu_i}{\beta} \right|.$$

Combining above two equations together, we have meanwhile that

$$\operatorname{sgn}(x_i^*) = \frac{\operatorname{sgn}(x_i^*)|x_i^*| + \operatorname{sgn}(x_i)/\beta}{|x_i^*| + 1/\beta} = \frac{x_i^* + \operatorname{sgn}(x_i)/\beta}{|x_i^*| + 1/\beta} = \frac{y_i - \nu_i/\beta}{|y_i - \nu_i/\beta|} = \operatorname{sgn} \left(y_i - \frac{\nu_i}{\beta} \right).$$

Hence,

$$x_i^* = |x_i^*| \frac{x_i^*}{|x_i^*|} = |x_i^*| \frac{(y_i - \nu_i/\beta)}{|y_i - \nu_i/\beta|} = \left(\left| y_i - \frac{\nu_i}{\beta} \right| - \frac{1}{\beta} \right) \operatorname{sgn} \left(y_i - \frac{\nu_i}{\beta} \right). \quad (2.17)$$

Furthermore, according to (2.16), $x_i^* = 0$ if and only if

$$\left|y_i - \frac{\nu_i}{\beta}\right| \leq \frac{1}{\beta}.$$

Coupling this to (2.17), we instantly conclude that

$$x_i^* = \max \left\{ \left|y_i - \frac{\nu_i}{\beta}\right| - \frac{1}{\beta}, 0 \right\} \operatorname{sgn}\left(y_i - \frac{\nu_i}{\beta}\right),$$

It can be written in a vector form; i.e.,

$$x^* = \max \left\{ \left|y - \frac{\nu}{\beta}\right| - \frac{1}{\beta}, 0 \right\} \operatorname{sgn}\left(y - \frac{\nu}{\beta}\right).$$

□

In light of Lemma 2, w -subproblem (2.13) can be explicitly solved when $\|\cdot\|$ is 1-norm; i.e.,

$$w_{i,k+1} = \max \left\{ \left|D_i u_k - \frac{\nu_i}{\beta_i}\right| - \frac{1}{\beta_i}, 0 \right\} \operatorname{sgn}\left(D_i u_k - \frac{\nu_i}{\beta_i}\right). \quad (2.18)$$

Lemma 3. For $x \in \mathbb{R}^p$, the subdifferential of $f(x) \triangleq \|x\|_2$ is

$$\partial f(x) = \begin{cases} x/\|x\|_2, & \text{if } x \neq 0; \\ \{h : \|h\|_2 \leq 1, h \in \mathbb{R}^p\}, & \text{otherwise.} \end{cases}$$

The proof of Lemma 3 is elementary and can be found in [18].

Lemma 4. For given $\beta > 0$ and $\nu, y \in \mathbb{R}^q$, the minimizer of

$$\min_x \|x\|_2 - \nu^T(y - x) + \frac{\beta}{2} \|y - x\|_2^2 \quad (2.19)$$

is given by the 2D shrinkage-like formula

$$x^* = \max \left\{ \left\| y - \frac{\nu}{\beta} \right\|_2 - \frac{1}{\beta}, 0 \right\} \frac{(y - \nu/\beta)}{\|y - \nu/\beta\|_2}, \quad (2.20)$$

where it follows the convention $0 \cdot (0/0) = 0$.

Proof. We use $\|\cdot\|$ for $\|\cdot\|_2$ for simplicity in this proof. Similar statements to Lemma 2 lead to the fact that there exists at least one minimizer x^* for (2.19) and the subdifferential of the objective function at this minimizer should contain the origin.

In light of Lemma 3, x^* must satisfy

$$\begin{cases} x^*/\|x^*\| + \beta(x^* - y) + \nu = 0 & \text{if } x^* \neq 0; \\ \|\nu - \beta y\| \leq 1 & \text{otherwise.} \end{cases} \quad (2.21)$$

If $x^* \neq 0$, it holds

$$x^* + x^*/(\beta\|x^*\|) = y - \frac{\nu}{\beta}, \quad (2.22)$$

which leads to

$$\|x^*\| + \frac{1}{\beta} = \left\| y - \frac{\nu}{\beta} \right\|. \quad (2.23)$$

Dividing (2.22) by (2.23), we obtain that

$$\frac{x^*}{\|x^*\|} = \frac{x^* + x^*/(\beta\|x^*\|)}{\|x^*\| + 1/\beta} = \frac{y - \nu/\beta}{\|y - \nu/\beta\|}.$$

This relation and (2.23) imply that

$$x^* = \|x^*\| \frac{x^*}{\|x^*\|} = \|x^*\| \frac{y - \nu/\beta}{\|y - \nu/\beta\|} = \left(\|y - \frac{\nu}{\beta}\| - \frac{1}{\beta} \right) \frac{y - \nu/\beta}{\|y - \nu/\beta\|}. \quad (2.24)$$

Moreover, $x^* = 0$ if and only if

$$\|y - \frac{\nu}{\beta}\| \leq \frac{1}{\beta}$$

according to (2.21). Combining this with (2.24), we instantly achieve

$$x^* = \max \left\{ \|y - \frac{\nu}{\beta}\| - \frac{1}{\beta}, 0 \right\} \frac{(y - \nu/\beta)}{\|y - \nu/\beta\|}.$$

□

In light of Lemma 4, the closed form solution of w -subproblem (2.13) can also be given out explicitly when $\|\cdot\|$ is 2-norm; i.e.,

$$w_{i,k+1} = \max \left\{ \|D_i u_k - \frac{\nu_i}{\beta_i}\| - \frac{1}{\beta_i}, 0 \right\} \frac{(D_i u_k - \nu_i/\beta_i)}{\|D_i u_k - \nu_i/\beta_i\|}, \quad (2.25)$$

where $0 \cdot (0/0) = 0$ is followed here as well.

Therefore, the w -subproblem derived from the process of minimizing either anisotropic or isotropic TV model can be solved exactly. For convenience, updating formulas (2.18) and (2.25) are uniformly denoted as

$$w_{i,k+1} = \text{shrike}(D_i u_k; \nu_i, \beta_i), \quad (2.26)$$

which is also the minimizer of w -subproblem (2.13). Here, the operator “shrike” is named from the abbreviation of “shrinkage-like formulas”. The complexity of (2.26) primarily focuses on computing the finite differences, which are almost negligible

compared with the same-size matrix-vector multiplications.

2.3.2 One-step Steepest Descent Scheme

In addition, with the aid of $w_{i,k+1}$, u_{k+1} can be achieved by solving

$$\begin{aligned} \min_u \mathcal{L}_A(w_{i,k+1}, u) &= \sum_i (\|w_{i,k+1}\| - \nu_i^T (D_i u - w_{i,k+1}) + \frac{\beta_i}{2} \|D_i u - w_{i,k+1}\|_2^2) \\ &\quad - \lambda^T (A u - b) + \frac{\mu}{2} \|A u - b\|_2^2, \end{aligned}$$

which is equivalent to solve the so-called “ u -subproblem”

$$\begin{aligned} \min_u \mathcal{Q}_k(u) &\triangleq \sum_i (-\nu_i^T (D_i u - w_{i,k+1}) + \frac{\beta_i}{2} \|D_i u - w_{i,k+1}\|_2^2) \\ &\quad - \lambda^T (A u - b) + \frac{\mu}{2} \|A u - b\|_2^2. \end{aligned} \quad (2.27)$$

Clearly, $\mathcal{Q}_k(u)$ is a quadratic function and its gradient is

$$d_k(u) = \sum_i (\beta_i D_i^T (-D_i u - w_{i,k+1}) - D_i^T \nu_i) + \mu A^T (A u - b) - A^T \lambda. \quad (2.28)$$

Forcing $d_k(u) = 0$ gives us the exact minimizer of $\mathcal{Q}_k(u)$

$$u_{k+1}^* = \left(\sum_i \beta_i D_i^T D_i + \mu A^T A \right)^+ \left(\sum_i (D_i^T \nu_i + \beta_i D_i^T w_{i,k+1}) + A^T \lambda + \mu A^T b \right) \quad (2.29)$$

where M^+ stands for the Moore-Penrose pseudoinverse of matrix M . Theoretically, it is ideal to accept the exact minimizer as the solution of the u -subproblem (2.27). However, computing the inverse or pseudoinverse at each iteration is too costly to implement numerically. Therefore, an iterative method is highly desirable.

The steepest descent method is able to solve (2.27) iteratively by applying recur-

rence formula

$$\tilde{u} = u - \alpha d,$$

where d is the gradient direction of the objective function. Each iteration of the steepest descent method demands updating the gradient direction, whose complexity is principally two matrix-vector multiplications on computing $A^T A u$. Thus, n -step steepest descent to obtain the minimizer of $\mathcal{Q}_k(u)$ requires $2n$ matrix-vector multiplications at least. For large-scale problems, it is still too costly to be an efficient algorithm. In fact, the augmented Lagrangian function (2.8) is expected to be minimized by solving w -subproblem (2.13) and u -subproblem (2.27) alternately. Therefore, solving the u -subproblem accurately at each sweep may be unnecessary. Instead of adopting multi-step steepest descent, we only take one aggressive step starting off with u_k , the approximate minimizer of $\mathcal{Q}_{k-1}(u)$, and accept the iterate as the roughly approximate minimizer of $\mathcal{Q}_k(u)$ (named one-step steepest descent method); i.e.,

$$u_{k+1} = u_k - \alpha_k d_k, \tag{2.30}$$

where $d_k \triangleq d_k(u_k)$ for simplicity.

The only remaining issue is how to choose α_k aggressively. Barzilai and Borwein [82] suggested an aggressive manner to choose step length for the steepest descent method, which is called the BB step or BB method. As can be seen, the BB step utilizes the previous two iterates and achieves the superlinear convergence [82, 83]. Surprisingly, Barzilai and Borwein's analysis also indicates that the convergence rate is even faster as the problem is more ill-conditioned. However, the one-step steepest descent is not able to offer two iterates, so we provide u_k and u_{k-1} by way of required

iterates to derive the BB-like step, which leads to

$$\alpha_k = \frac{s_k^T s_k}{s_k^T y_k}, \quad (2.31)$$

or

$$\alpha_k = \frac{s_k^T y_k}{y_k^T y_k}, \quad (2.32)$$

where $s_k = u_k - u_{k-1}$ and $y_k = d_k(u_k) - d_k(u_{k-1})$.

To validate the BB-like step, a nonmonotone line search algorithm (NLSA) advanced by Zhang and Hager [84] is integrated. They modified the scheme of Grippo, Lampariello, and Lucidi [85] on nonmonotone line search and demonstrated their new algorithm was generally superior to the traditional one [85] according to a large number of numerical experiments. From iteration to iteration, NLSA requires checking the nonmonotone Armijo condition, which is

$$\mathcal{Q}_k(u_k - \alpha_k d_k) \leq C_k - \delta \alpha_k d_k^T d_k. \quad (2.33)$$

where C_k is recursively set by an average of function values; i.e.,

$$\begin{aligned} P_{k+1} &= \eta P_k + 1, \\ C_{k+1} &= (\eta P_k C_k + \mathcal{Q}_k(u_{k+1})) / P_{k+1}, \end{aligned} \quad (2.34)$$

and δ and η are chosen between 0 and 1.

So far all issues in the process of handling the subproblem have been settled. In light of all derivations above, the new algorithm to minimize the augmented Lagrangian function (2.8) is stated as follows:

Algorithm 2 (Alternating Minimization Scheme).

Initialize $0 < \delta, \rho, \eta < 1$ and starting points $w_{i,0}, u_0$;

Set $Q_0 = 1$ and $C_0 = \mathcal{L}_A(w_{i,0}, u_0)$;

While *inner stopping criteria unsatisfied* **Do**

Compute $w_{i,k+1}$ based on shrinkage-like formula (2.26);

Set α_k through BB-like formula (2.31);

While *nonmonotone Armijo condition (2.33) unsatisfied* **Do**

Backtrack $\alpha_k = \rho\alpha_k$;

End Do

Compute u_{k+1} by one-step steepest descent method (2.30);

Set C_{k+1} according to (2.34);

End Do

About selecting the inner stopping criteria, there are at least two optional ways:

- $\|\nabla\mathcal{L}_A(w_{i,k}, u_k)\|_2$ is sufficiently small;
- relative change $\|u_{k+1} - u_k\|_2$ is sufficiently small.

2.4 Overall Algorithm and Extensions

By means of a combination of Augmented Lagrangian Method and Alternating Minimization Scheme, the TV model (2.1) can be efficiently optimized. More precisely, the new TV solver TVAL3 implements the following algorithmic framework:

Algorithm 3 (TVAL3 Scheme).

Initialize $\nu_i^0, \beta_i^0, \lambda^0, \mu^0$, and starting points w_i^0, u^0 for all i ;

While *outer stopping criteria unsatisfied* **Do**

Set $w_{i,0}^{k+1} = w^k$ and $u_0^{k+1} = u^k$;

Find minimizers w_i^{k+1} and u^{k+1} of the augmented Lagrangian function (2.8)

by means of Algorithm 2, starting from $w_{i,0}^{k+1}$ and u_0^{k+1} ;

Update multipliers using (2.9) to attain $\nu_i^{k+1}, \lambda^{k+1}$;

Choose new penalty parameters $\beta_i^{k+1} \geq \beta_i^k$ and $\mu^{k+1} \geq \mu^k$;

End Do

Similar to the inner stopping criteria, there are also at least two ways to choose the outer stopping criteria:

- optimality conditions of (2.7) are approximately achieved;
- relative change $\|u^{k+1} - u^k\|_2$ is sufficiently small.

This algorithmic framework is flexible; in fact, it could be extended to some other TV models with various constraints in the field of compressive sensing. For instance, For the TV model with nonnegativity constraints,

$$\min_u \sum_i \|D_i u\|, \quad \text{s.t. } Au = b \text{ and } u \geq 0, \quad (2.35)$$

we take one step of the projected gradient method [86] instead of the steepest descent method while updating u . Except for this modification, all the other details in Algorithm 3 remain the same to deal with the TV model with nonnegativity constraints (2.35).

With slight modifications on updating formulas, but following the same derivations, Algorithm 3 can also be used to recover complex signals or images, which means solving (2.1) under $u \in \mathcal{C}^n$ or $u \in \mathcal{C}^{s \times t}$ with $s \cdot t = n$ and $A \in \mathcal{C}^{m \times n}$ with $m < n$.

A new solver TVAL3—a main contribution of this thesis—implementing algorithms grounded on the TVAL3 scheme has been published at the following URL:

<http://www.caam.rice.edu/~optimization/L1/TVAL3/>.

The theoretical conclusions on convergence or convergence rate have not yet been thoroughly investigated, even though solid numerical evidence reveals that these algorithms do converge. Theoretical investigations on convergence would be part of my future research. In the fourth chapter, the results of a large number of numerical experiments, which aim at 1D and 2D, noisy and noise-free, real and complex, and regular and SPC signals or images (generated by the single pixel camera), will strongly indicate the convergence of the TVAL3 scheme in practice. Before that, a type of measurement matrices with special structure which could significantly accelerate the TVAL3 scheme, will be well studied in the following chapter.

Chapter 3

Fast Walsh Hadamard Transform

In this chapter, a type of structured measurement matrices, which is adopted by the single pixel camera, is taken into account to accelerate the **TVAL3** scheme for CS problems. As proposed in Chapter 2, Algorithm 3 is essentially based on the following two recursive formulas

$$\begin{aligned}w_{i,k+1} &= \text{shrike}(D_i u_k; \nu_i, \beta_i), \\u_{k+1} &= u_k - \alpha_k d_k,\end{aligned}$$

where

$$d_k = \sum_i (\beta_i D_i^T (-D_i u_k - w_{i,k+1}) - D_i^T \nu_i) + \mu A^T (A u_k - b) - A^T \lambda.$$

Because computing the finite difference is much less expensive than matrix-vector multiplication in MATLAB, two matrix-vector multiplications Au^k and $A^T(Au^k - b)$ dominate the running time at each iteration. Specifically, assuming that the size of matrix A is $m \times n$ and that computing Ax takes $c(m, n)$, then the running time of the

new algorithm is briefly $c(m, n) \times p$ where p is the number of total iterations. For the fixed image size and recovery percentage (i.e. fixed m and n), obviously two ways are available to accelerate the algorithm: making p smaller or making $c(m, n)$ smaller.

Making p smaller requires modification of the algorithm, and even the core part, to improve the convergence rate. This is a difficult task, especially for a completed algorithm. Perhaps the adjustment of parameters would make some differences or even some improvements, but the optimal parameters are hard to find and vary from case to case. It can be considered as an independent and open research topic. Making p smaller is correspondingly easier. It requires a fast way to handle the matrix-vector multiplication. Some structured measurements, originated from special transforms such as Fourier, Cosine, or Walsh Hadamard transforms, are able to handle the fast computation of matrix-vector multiplication.

The measurement matrix A generated by the digital micro-mirror device (DMD) of the single pixel camera is programmed as a permuted Walsh Hadamard matrix. In fact, during the hardware implementation, the matrix entries -1 and 1 are shifted to 0 and 1 so that DMD can correctly recognize. It is essential to explore the Walsh Hadamard transform and find a fast way to implement it. This chapter therefore starts with introducing the basic concept of the Hadamard Matrix.

3.1 Hadamard Matrix

The Hadamard matrix or transform is named for the French mathematician Jacques Solomon Hadamard, the German-American mathematician Hans Adolph Rademacher, and the American mathematician Joseph Leonard Walsh. It belongs to a generalized class of Fourier transforms and performs an orthogonal, symmetric, involutorial, linear operation on 2^k real numbers.

The Hadamard matrix of dimension 2^k for $k \in \mathbb{N}$ are given by the recursive formula

$$H_0 = [1],$$

$$H_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

and in general,

$$H_k = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix}.$$

According to this formula, for instance,

$$H_3 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}.$$

This is also known as the Hadamard-ordered Walsh Hadamard matrix. There are also other orders, such as sequency order, dyadic order, and so forth. Different orders can be achieved by re-ordering the rows of the Hadamard matrix defined above. Walsh Hadamard matrices in various orders have recently received increasing attention due

to their broad applications in the field of engineering. The hadamard and dyadic orders are more appropriate for applications involving a double transform (time-space-time) such as logical autocorrelation and convolution [47]. The sequency order can be applied to sequency filters, sequency power spectra, and so forth. In the single pixel camera, each pattern of DMD corresponds to a row of the permuted sequency ordered Walsh Hadamard matrix after shifting entries from -1 and 1 to 0 and 1 . Hence the sequency order is the main focus in this chapter.

To convert a given sequency integer number s into the corresponding index number k in Hadamard order, one needs the following steps [94]:

- Represent s in binary form:

$$s = (s_{n-1}s_{n-2} \dots s_0)_2 = \sum_{i=0}^{n-1} s_i 2^i.$$

- Transfer the binary form to Gray code[48]:

$$g_i = s_i \oplus s_{i+1} \quad i = 0, 1, \dots, n-1,$$

where \oplus stands for *exclusive or* and $s_n = 0$.

Specifically,

$$1 \oplus 1 = 0 \oplus 0 = 0; \quad 1 \oplus 0 = 0 \oplus 1 = 1.$$

- Reverse g_i 's bit to achieve k_i 's:

$$k_i = g_{n-1-i}.$$

For example, $n = 3$ we have

s	0	1	2	3	4	5	6	7
binary	000	001	010	011	100	101	110	111
Gray code	000	001	011	010	110	111	101	100
bit-reverse	000	100	110	010	011	111	101	001
k	0	4	6	2	3	7	5	1

Let $A(i)$ denote the $(i + 1)$ th row of matrix A . Based on the above form, define

$$W_3(i) = H_3(s(i));$$

i.e.,

$$\begin{aligned}
W_3 &= [H_3(0)^T \quad H_3(4)^T \quad H_3(6)^T \quad H_3(2)^T \quad H_3(3)^T \quad H_3(7)^T \quad H_3(5)^T \quad H_3(1)^T]^T \\
&= \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}.
\end{aligned}$$

W_3 is sequency-ordered Walsh Hadamard matrix.

Based on this process, $2^k \times 2^k$ sequency-ordered Walsh Hadamard matrix can be simply generated for any integer k .

To achieve the fast Walsh Hadamard transform, it is necessary to understand the so-called "Kronecker product", which will be discussed in the next section.

3.2 Kronecker Product and Fast Walsh Hadamard Transform

For any two matrices $A = [a_{ij}]_{p \times q}$ and $B = [b_{ij}]_{r \times l}$, the *Kronecker product* of these two matrices is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1q}B \\ a_{21}B & a_{22}B & \dots & a_{2q}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{p1}B & a_{p2}B & \dots & a_{pq}B \end{bmatrix}_{pr \times ql}$$

To study an essential property of the Kronecker product, I need to define two new operators *vec* and *mtx*. Specifically, *vec* is the operator that stacks the columns of a matrix to form a vector, and *mtx* separates the vector into several equal-length vectors and forms a matrix. The size of the reshaped vector or matrix depends on the size of matrices before and after it when computing matrix-matrix or matrix-vector multiplication to guarantee the success of computation. The following example and theorem would make this point more clear. Literally, *mtx* is the inverse operator of *vec*.

For example,

$$X = \begin{bmatrix} 1 & 2 \\ -1 & 4 \\ 6 & 7 \end{bmatrix},$$

then

$$x = \mathbf{vec}(X) = \begin{bmatrix} 1 \\ -1 \\ 6 \\ 2 \\ 4 \\ 7 \end{bmatrix} \quad \text{and} \quad \mathbf{mtx}(x) = X.$$

With the aid of two new operators, the following well-known theorem can be concluded:

Theorem 3 (the Basic KP theorem). *Matrix $A \in \mathbb{R}^{n \times m}$ is constructed by the Kronecker product formula*

$$A = A_1 \otimes A_2,$$

where $A_1 \in \mathcal{R}^{(m/p) \times (n/q)}$ and $A_2 \in \mathcal{R}^{p \times q}$. m and n are chosen to satisfy that m and n are divisible by p and q , respectively. Then matrix-vector multiplication can be computed by

$$\begin{aligned} Ax &= \mathbf{vec}(A_2 \mathbf{mtx}(x) A_1^T), \\ A^T y &= \mathbf{vec}(A_2^T \mathbf{mtx}(y) A_1). \end{aligned}$$

Proof. Define $s = m/p$, $t = n/q$, and $A_1 = (a_{ij})_{s \times t}$.

Furthermore, denote $x = [x_1, \dots, x_t]^T$, then $\mathbf{mtx}(x) = [x_1, \dots, x_t]$.

$$Ax = (A_1 \otimes A_2)x$$

By the definition of Kronecker product,

$$= \begin{bmatrix} a_{11}A_2 & a_{12}A_2 & \dots & a_{1t}A_2 \\ a_{21}A_2 & a_{22}A_2 & \dots & a_{2t}A_2 \\ \vdots & \vdots & \vdots & \vdots \\ a_{s1}A_2 & a_{s2}A_2 & \dots & a_{st}A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_t \end{bmatrix}$$

According to the matrix-vector multiplication,

$$= \begin{bmatrix} a_{11}A_2x_1 + a_{12}A_2x_2 + \dots + a_{1t}A_2x_t \\ a_{21}A_2x_1 + a_{22}A_2x_2 + \dots + a_{2t}A_2x_t \\ \vdots \\ a_{s1}A_2x_1 + a_{s2}A_2x_2 + \dots + a_{st}A_2x_t \end{bmatrix}$$

By the definition of two new operators,

$$= \mathbf{vec}([a_{11}A_2x_1 + a_{12}A_2x_2 + \dots + a_{1t}A_2x_t, \dots, a_{s1}A_2x_1 + a_{s2}A_2x_2 + \dots + a_{st}A_2x_t])$$

By the simple reorganization,

$$= \mathbf{vec}([A_2[x_1, \dots, x_t][a_{11}, \dots, a_{1t}]^T, \dots, A_2[x_1, \dots, x_t][a_{s1}, \dots, a_{st}]^T])$$

Rewriting in the matrix form,

$$\begin{aligned}
&= \mathbf{vec} \left(A_2 [x_1, \dots, x_t] \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1t} \\ a_{21} & a_{22} & \dots & a_{2t} \\ \vdots & \vdots & \vdots & \vdots \\ a_{s1} & a_{s2} & \dots & a_{st} \end{bmatrix}^T \right) \\
&= \mathbf{vec}(A_2 X A_1^T).
\end{aligned}$$

The same argument can prove

$$A^T y = \mathbf{vec}(A_2^T \mathbf{mtx}(y) A_1).$$

□

Using the Kronecker product, the formula (3.1) can be rewritten as

$$H_k = H_1 \otimes H_{k-1}.$$

For any given vector x with the length of 2^k , denote $x = [x_1^T \quad x_2^T]^T$, where x_1 and x_2 are of equal size. The Hadamard-ordered Walsh Hadamard transform (WHT_h) can be written as

$$H_k x = (H_1 \otimes H_{k-1}) x.$$

Due to the Basic KP theorem, it follows

$$\begin{aligned}
H_k x &= \mathbf{vec}(H_{k-1} \mathbf{mtx}(x) H_1^T) \\
&= \mathbf{vec}(H_{k-1} [x_1 \quad x_2] H_1^T) \\
&= \mathbf{vec}([H_{k-1} x_1 \quad H_{k-1} x_2] H_1^T) \\
&= \frac{1}{\sqrt{2}} \mathbf{vec} \left([H_{k-1} x_1 \quad H_{k-1} x_2] \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right) \\
&= \frac{1}{\sqrt{2}} \mathbf{vec}([H_{k-1} x_1 + H_{k-1} x_2 \quad H_{k-1} x_1 - H_{k-1} x_2]) \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} H_{k-1} x_1 + H_{k-1} x_2 \\ H_{k-1} x_1 - H_{k-1} x_2 \end{bmatrix}. \tag{3.1}
\end{aligned}$$

A naive implementation of the WHT_h would have a computational complexity of $O(N^2)$, but the fast WHT_h implementation according to recursive formula (3.1) requires only $O(N \log N)$. Notice that only additions and subtractions are involved while implementing the fast WHT_h . Sequence-ordered Walsh Hadamard transform (WHT_s) is directly obtained by carrying out the fast WHT_h as above, and then rearranging the outputs by bit-reverse and Gray code conversion.

I will show some comparison results in the next section to illustrate how fast the newly implemented Walsh Hadamard transform is based on the running time.

3.3 Comparisons

I implemented the fast Walsh Hadamard transform in C++ and then compiled and linked it into a shared library called a binary *MEX*-file from MATLAB software. The fast Walsh Hadamard transform was also carried out since the version of MATLAB R2008b, which is known as function *fwht* and its inverse function *ifwht*. The following

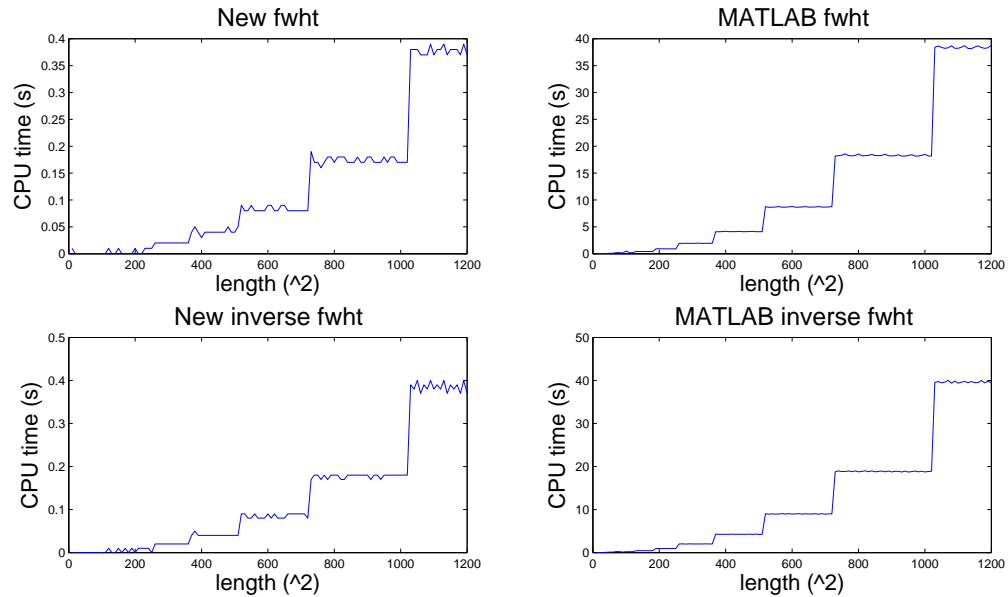


Figure 3.1: Running time comparison between newly implemented FWHT and MATLAB function *fwht*. Clearly, newly implemented FWHT is around 100 times faster than FWHT provided by MATLAB.

experiments compare the newly implemented FWHT and its inverse with MATLAB functions. All experiments were performed on a Lenovo X301 laptop running Windows XP and MATLAB R2009a (32-bit) and equipped with a 1.4GHz Intel Core 2 Duo SU9400 and 2GB of DDR3 memory.

Figure 3.1 illustrates that my newly implemented code to compute the fast WHT is much faster than MATLAB function *fwht* and *fwht* (around 1/100 running time on average), and Figure 3.2 illustrates that the fast WHT can be even faster than the fast Fourier transform (around 1/2 running time on average), which clearly shows the efficiency of the newly implemented fast WHT.

Obviously, computing the matrix-vector multiplication in such a fast way can accelerate the TVAL3 scheme. More numerical results to demonstrate the efficiency and robustness of the corresponding algorithms will be shown in next chapter.

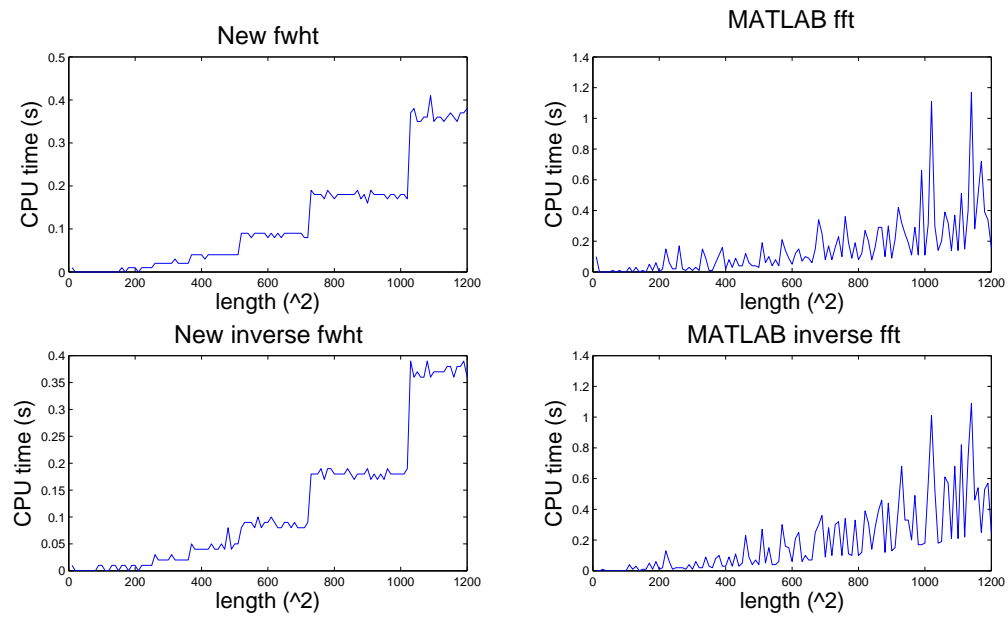


Figure 3.2: Running time comparison between newly implemented FWHT and MATLAB function *fft*. Clearly, newly implemented FWHT is even faster than *fft* provided by MATLAB, which is nearly the most efficient transform implemented by MATLAB.

Chapter 4

Numerical Results and Discussions

In this chapter, the effectiveness and efficiency of TVAL3 on image reconstruction is demonstrated by reporting the procedure and results of a large number of numerical experiments. TVAL3 is compared with other state-of-the-art TV solvers, as well as ℓ_1 solvers to validate its advantages. All experiments fall under two categories: reconstructing test images obtained from public domain and recovering images from real data generated by the single pixel camera (SPC) or by related techniques. The true solutions can be predefined for the first category whereas that is unlikely for the second category. That means true images are rarely available for reference while recovering real data. However, the single pixel camera is the main application of TVAL3 and its data is much closer to practical applications. Thus, simulating results based on SPC data or other real data are more indicative and convincing.

4.1 State-of-the-art Solvers and Test Platform

TV solvers have been introduced in Section 1.3. Since SOCP [19] is much slower than others and RecPF [36] is restricted to partial Fourier measurements only, these two

solvers will be omitted from the comparison. In other words, comparisons are primarily made among TVAL3 (version beta2.1), TwIST (version 1.0) [57, 58], NESTA (version 1.0) [56], and ℓ_1 -Magic (version 1.1) [3, 2, 1]. It is noteworthy that there are two available reconstruction codes in the current version of NESTA—*NESTA.m* and *NESTA_UP.m*. The only difference is *NESTA.m* requires $A^T A$ to be an orthogonal projector but *NESTA_UP.m* has no particular requirements on measurement matrix A . Therefore, *NESTA_UP.m* is adopted whenever NESTA is involved in any numerical experiment. Additionally, the two state-of-the-art ℓ_1 solvers, FPC (version 2.0) [17] and YALL1 (version beta5.0) [59], are involved in some experiments to indicate the merits of TV solvers compared to ℓ_1 solvers. FPC and YALL1 are among the best solvers for ℓ_1 minimization in terms of both speed and accuracy.

While running TVAL3, we uniformly set parameters $\delta = 1.e - 5$, $\rho = .6$, and $\eta = .9995$ presented in Algorithm 2, and $\nu_i^0 = 0$, $\lambda^0 = 0$, $u^0 = A^T b$, $w_i^0 = \mathit{shike}(D_i u_0; \nu_i^0, \beta_i^0)$ presented in Algorithm 3. Additionally, penalty parameters β_i^k and μ^k are chosen without continuation but kept constant equal to the initial values β_i^0 and μ^0 , respectively. The values of β_i^0 , μ^0 , and tolerance might vary according to distinct noise level and required accuracy.

In an effort to make the comparisons fair, for other tested solvers mentioned above, different choices of parameters have always been tried and at the end we pick out the ones that provide the best performance measured by recovery quality and running time.

All experiments were performed on a Lenovo X301 laptop running Windows XP and MATLAB R2009a (32-bit) and equipped with a 1.4GHz Intel Core 2 Duo SU9400 and 2GB of DDR3 memory.

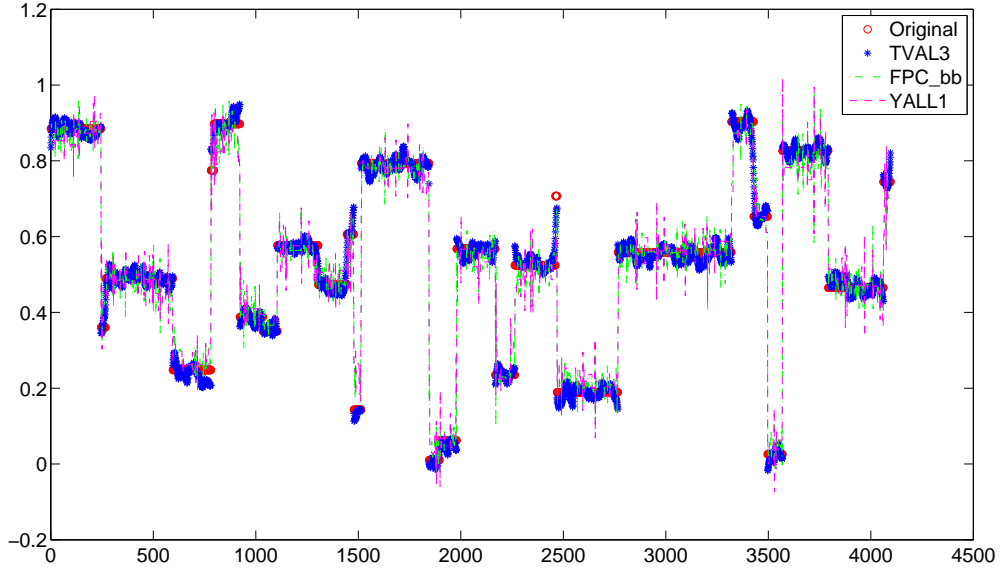


Figure 4.1: Reconstructed 1D staircase signal from 20% measurements. The noise level is 4%. Relative errors recovered by TVAL3, FPC_bb, and YALL1 are 3.31%, 6.37%, and 7.41%, and running times are 2.61s, 4.17s, and 2.62s, respectively.

4.2 Comparisons Based on Synthetic Data

In this section, the test sets cover 1D staircase signals, 2D Shepp-Logan phantom images, and the 2D MR brain image, with various sampling ratios. In each test, the observation f is generated by firstly stacking the columns of the tested image to form a vector and then applying the fast transform or general random matrix to it. The additive Gaussian noise on f has mean 0 and standard deviation 1 in all tests. In MATLAB, the noisy observation is explicitly given by

$$f = f + \sigma \cdot \text{mean}(\text{abs}(f)) \cdot \text{randn}(m, 1), \quad (4.1)$$

where σ represents the noise level and m represents the length of f .

Let us begin with recovering 1D staircase signals. In test 1 (corresponding to

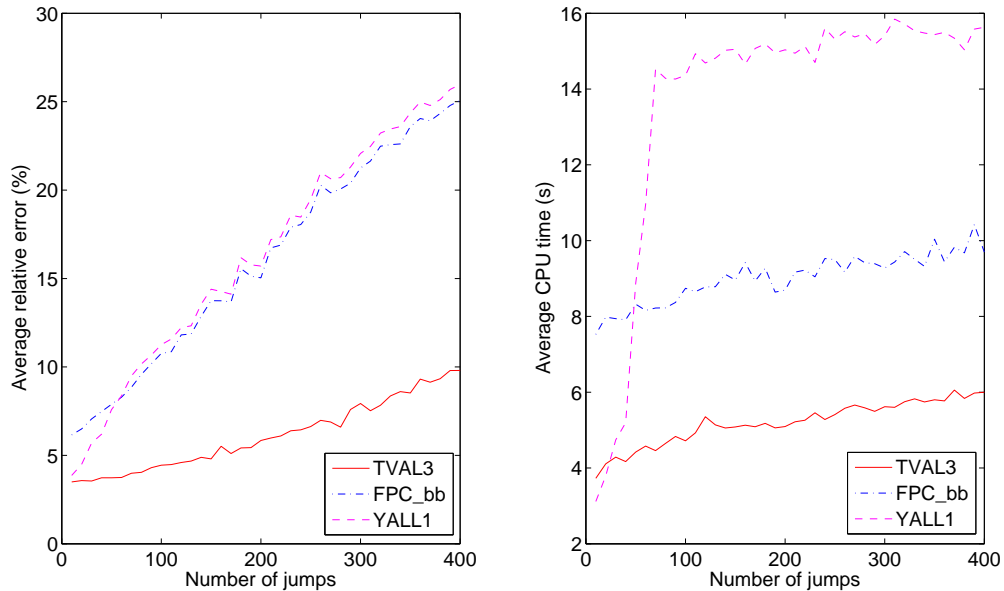


Figure 4.2: Recoverability for 1D staircase signals. The measurement rate is 40% and the noise level is 8%. **Left:** average relative error. **Right:** average running time. Relative error and running time are measured simultaneously with the growth of the number of jumps.

Figure 4.1), the length of the tested signal is 4096 with 27 jumps, the measurement matrix is Gaussian random matrix whose measurement rate is 20%, and the noise level is 4%. The current versions of all the other TV solvers except TVAL3 can only reconstruct 2D square images, although the methods behind some of these solvers can be extended to reconstruct non-square images. Therefore, TVAL3 is compared with the two ℓ_1 solvers—FPC_bb (FPC with Barzilai-Borwein steps) and YALL1. Since the signal is dense, it is sparsified by the Haar wavelet before FPC_bb or YALL1 is applied.

The parameters are set as default except assigning $opts.mu = 8$, $opts.beta = 8$, and $opts.tol = 1e - 3$ for TVAL3; assigning $opts.tol = 1e - 2$ for FPC_bb; assigning $opts.nu = 35$ and $opts.tol = 5e - 3$ for YALL1. Since the stopping criteria vary from solver to solver, we used different tolerance values for different solvers to achieve a fair

comparison. The guiding principle here is either to make running time approximately equal while comparing quality or the other way around. If the shorter running time and higher accuracy can be reached at the same time for one solver, it is also favorable for a fair comparison. As mentioned before, these parameters were chosen after multi-trials to provide the best observed results.

Figure 4.1 indicates that the new TV solver TVAL3 achieves higher accuracy within shorter running than the two ℓ_1 solvers, and the signal recovered by TVAL3 is less oscillatory.

The above statements are again validated by test 2 (corresponding to Figure 4.2). Fixing the length of 1D staircase signals to 4096, measurement rate of Gaussian random matrix to 40%, and noise level to 8%, we run the test when the number of jumps is 10, 20, 30, \dots , 400 respectively. We take 5 trials at each testing point and plot the average relative error and running time with respect to the number of jumps. The parameters of three solvers are set exactly the same as mentioned in test 1.

Figure 4.2 clearly demonstrates that relative error generated by TVAL3 increases much slower than relative error generated by either of the two ℓ_1 solvers with the increase in the number of jumps. Meanwhile, the running time of TVAL3 is much less than either of the two ℓ_1 solvers when the number of jumps is more than 30. When the number of jumps is relatively small (roughly less than 30 in this case), which correlates with the very sparse Haar wavelet coefficients, YALL1 becomes very efficient. Generally speaking, the TV solver TVAL3 gives better recoverability and higher efficiency compared to ℓ_1 solvers, at least for 1D staircase signals.

A series of experiments on 2D images which compare among TV solvers are described as follows. Test 3 and 4 are on noise-free cases, while test 5 and 6 on noisy cases.

In test 3 (corresponding to Figure 4.3), a 64×64 phantom image is encoded by an

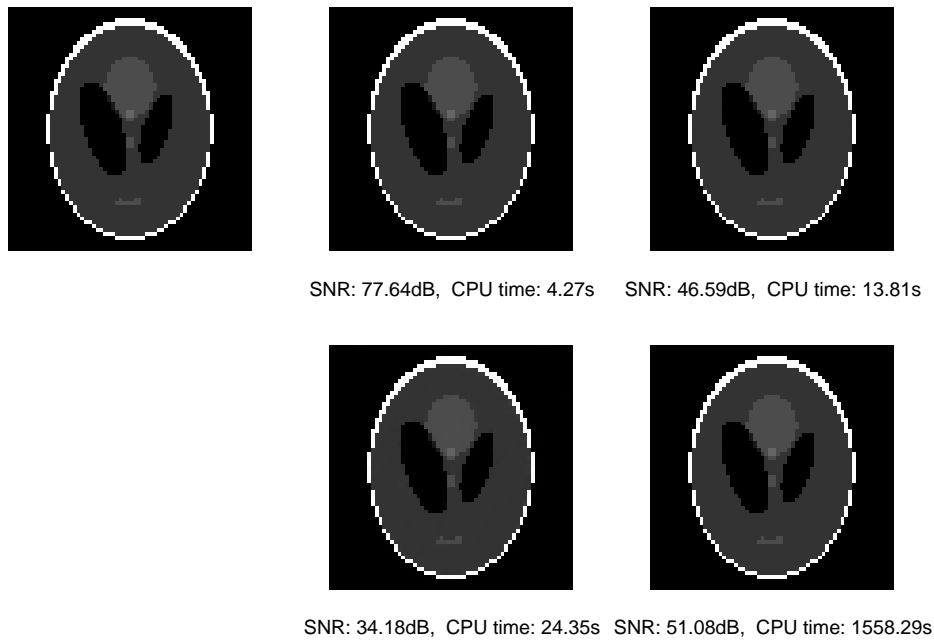


Figure 4.3: Recovered 64×64 phantom image from 30% orthonormal measurements without noise. **Top-left:** original image. **Top-middle:** reconstructed by TVAL3. **Top-right:** reconstructed by TwIST. **Bottom-middle:** reconstructed by NESTA. **Bottom-right:** reconstructed by ℓ_1 -Magic.

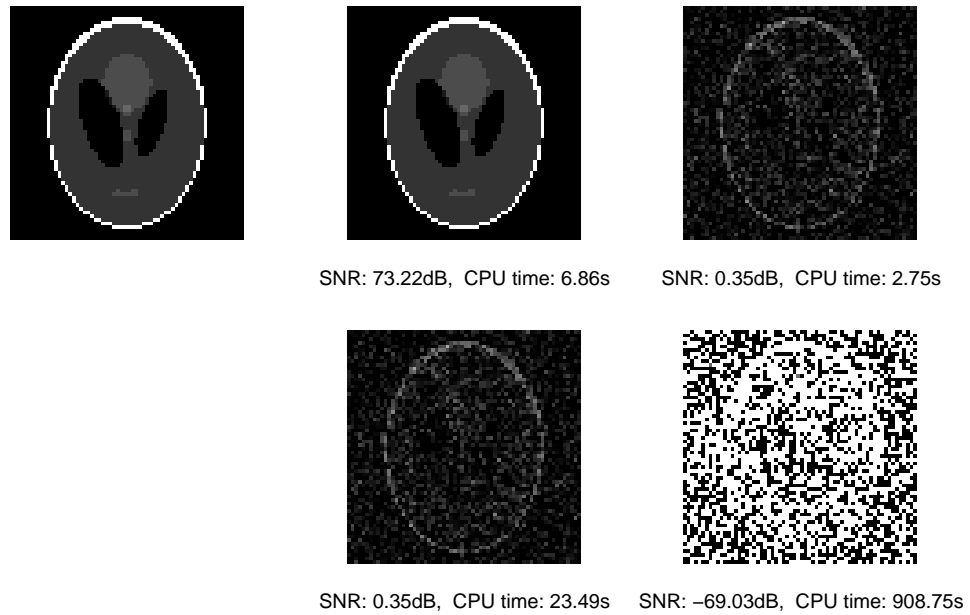


Figure 4.4: Recovered 64×64 phantom image from 30% non-orthonormal measurements without noise. **Top-left:** original image. **Top-middle:** reconstructed by TVAL3. **Top-right:** reconstructed by TwIST. **Bottom-middle:** reconstructed by NESTA. **Bottom-right:** reconstructed by ℓ_1 -Magic.

orthonormal random matrix generated by QR factorization from a Gaussian random matrix. The images are recovered by TVAL3, TwIST, NESTA, and ℓ_1 -Magic from 30% measurements but without the additive noise. The quality of recovered images is measured by the signal-to-noise ratio (SNR), which is defined as the power ratio between a signal and the background noise. Mathematically,

$$\text{SNR} = 20 \log_{10} \left(\frac{\|u_{ref} - \text{mean}(u_{ref})\mathbb{1}\|_F}{\|u_{cal} - u_{ref}\|_F} \right),$$

where u_{cal} and u_{ref} represent the recovered and original images respectively, $\mathbb{1}$ represents the matrix of all ones whose size is the same as u_{ref} , $\|\cdot\|_F$ calculates the Frobenius norm, and the operator **mean** calculates the mean value of all entries in a matrix.

The chosen parameter settings for this test after multi-trials are $opts.mu = 2^8$ and $opts.tol = 1e-4$ for TVAL3; $tau = 1/2000$ and $tolA = 1e-4$ for TwIST; $mu = 2e-3$, $Lambda = 1/2000$, $La = \|A\|_2^2$, and $opts.TOIVar = 1e-4$ for NESTA; $mu = 2$ and $lbtol = 1e-2$ for ℓ_1 -Magic. All other parameters are set up as default.

From Figure 4.3, we observe that TVAL3 achieves the highest-quality image (77.64dB) but requires the shortest running time (4.27 seconds). The second highest-quality image (51.08dB) is recovered by ℓ_1 -Magic at the expense of the unacceptable running time (1558.29 seconds). TwIST and NESTA attain relatively midium-quality images (around 46.59dB and 34.18dB respectively) within reasonable running times (13.81 and 24.35 seconds respectively). This test validates that TVAL3 is capable of high accuracy within an affordable running time for noise-free images.

Test 4 (corresponding to Figure 4.4) carries out the same experiment as test 3 except for replacing the orthonormal random matrix by the Gaussian random matrix as the measurement matrix. All the parameters are set exactly as described in test 3.

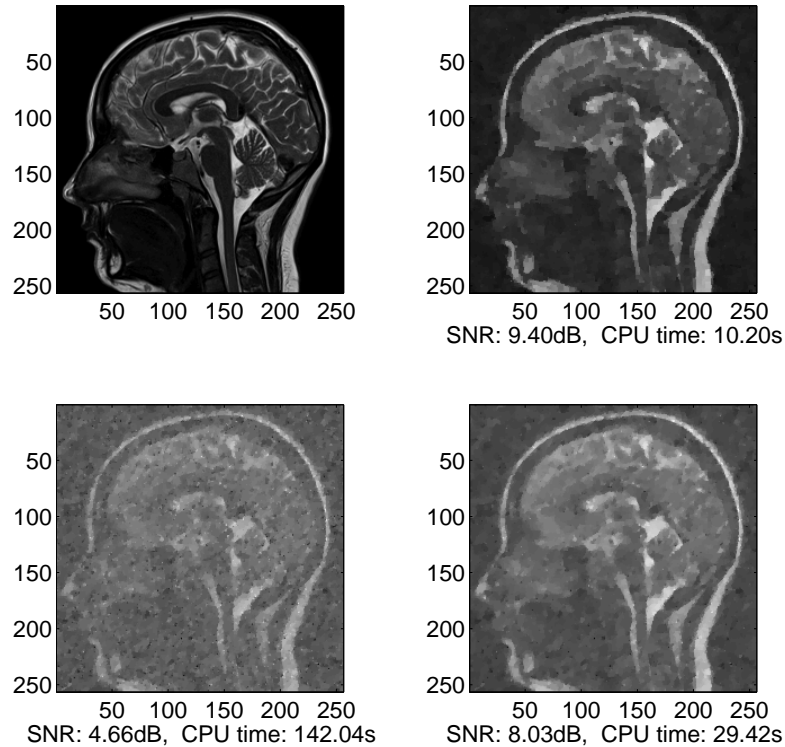


Figure 4.5: Recovered 256×256 MR brain image. Both the measurement rate and the noise level are 10%. **Top-left:** original image. **Top-right:** reconstructed by TVAL3. **Bottom-left:** reconstructed by TwIST. **Bottom-right:** reconstructed by NESTA.

It turns out that the non-orthonormal measurement matrix caused failures in TwIST, NESTA, and ℓ_1 -Magic, as evidenced in Figure 4.4. However, TVAL3 can still recover the phantom with high quality (73.22dB) within a reasonable time (6.86 seconds). This experiment attests to the versatility and robustness of TVAL3 with different measurement matrices.

In the next two tests, we focus on reconstructing a MR brain image to reveal the potential of TVAL3 in the field of medical imaging. Since ℓ_1 -Magic is hardly applicable to large-scale problems as shown in test 3 and 4, TVAL3 is only compared with TwIST and NESTA.

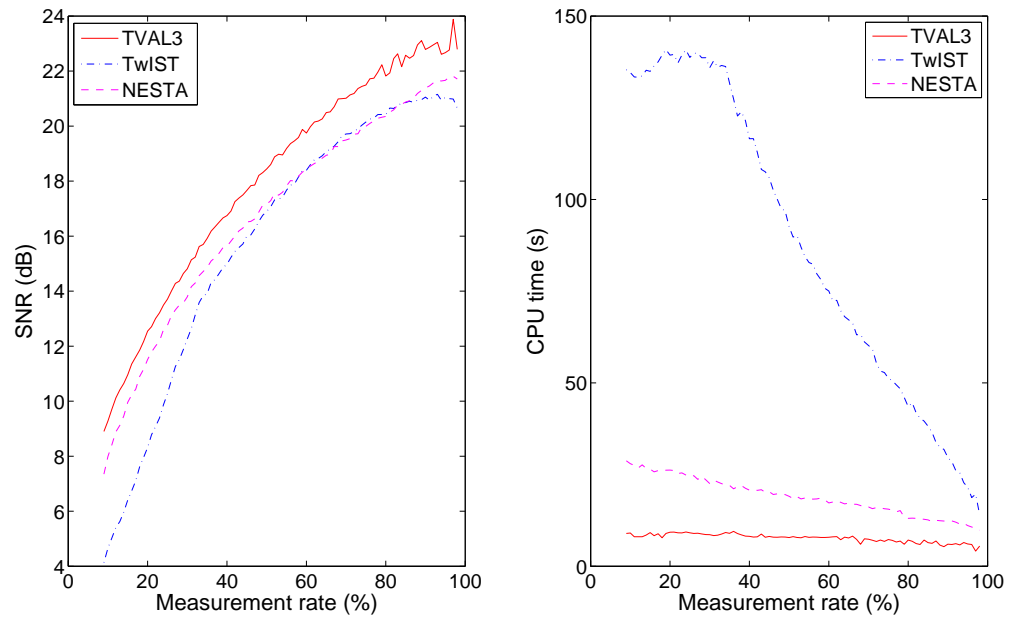


Figure 4.6: Recoverability for 256×256 MR brain image. The noise level is 10%. **Left:** average SNR. **Right:** average running time. SNR and running time are measured simultaneously with the growth of the measurement rate.

In test 5 (corresponding to Figure 4.5), a 256×256 MR brain image, which is more complex and harder to reconstruct than phantom images, is encoded by a permuted Walsh Hadamard matrix. The sequency-ordered Walsh Hadamard transform as described in Chapter 3 is performed here to shorten the running time for all solvers. In order to investigate the robustness, we try to push solvers to the limit by adding a lot of noise and using a small number of measurements. More precisely, noise level and measurement rate are both set to 10%.

The parameter settings are as follows: $opts.mu = 2^9$, $opts.beta = 2^8$, and $opts.tol = 4e - 3$ for TVAL3; $tau = 1/50$, $tolA = 1e - 3$, and $MaxiterA = 200$ for TwIST; $mu = 5e - 3$, $Lambda = 1/50$, $La = 1$, and $opts.TOIVar = 1e - 3$ for NESTA. Others are automatically set as default.

From Figure 4.5, we can only recognize the outline of the image recovered by TwIST even though the running time is longest. Nevertheless, the image recovered by either TVAL3 or NESTA keeps the rough sketch and some details of the original brain image. In comparison with NESTA, TVAL3 achieves better accuracy (higher SNR) in shorter running time statistically, and provides higher contrast visually. For example, some gyri in the image recovered by TVAL3 are still distinguishable but this is not the case in images recovered by either TwIST or NESTA. Furthermore, the image recovered by NESTA is still noisy while the image recovered by TVAL3 is much cleaner. This validates that TVAL3 is capable of better denoising effects while reconstructing than NESTA. This fact will be reconfirmed by those tests related to the single pixel camera in next section. Actually, this is an advantage when handling data with lots of noise, which will always be the case in practice.

Fixing noise level to 10%, test 6 (corresponding to Figure 4.6) repeats test 5 at 90 different measurement rates from 9% to 98%. Testing points are uniformly chosen and all parameters are set the same as in test 5.

Figure 4.6 indicates that TVAL3 always achieves the best quality (highest SNR) with the least running time among three TV solvers for that brain image. TwIST and NESTA attain close accuracy, but TwIST is much slower especially when the measurement rate is relatively low. These facts are consistent with what we discovered from Figure 4.5.

The above tests validate that TVAL3 is more efficient and robust in comparison to other TV solvers and even the two state-of-the-art ℓ_1 solvers when reconstructing some testing signals and images with Gaussian noise. More complicated data measured in practice are taken into account in next the section.

4.3 Comparisons Based on Measured Data

The following tests are focusing on the measured data, which were measured and provided by the Single-Pixel Camera Group from the ECE department of Rice University.

For measured data, the quality of recovered images is difficult to quantify due to the lack of true solutions. Thus, the following comparisons are more or less relying on visual effects. In each test of this section, the same tolerance is adopted for all the tested solvers, which means neither similar quality nor close running time among the recovered images. The reason for this is simply convenience. Test 7 focuses on reconstructing infrared data captured by the single pixel camera [44], and test 8 aims at recovering the signal using optical beam-induced current (OBIC) technique for laser-based failure-analysis [45].

The measurements which are adopted by the single pixel camera to decide the patterns of the digital micro-mirror device (DMD) are extracted from the permuted Walsh Hadamard matrix. This matrix can be efficiently performed by the sequency-

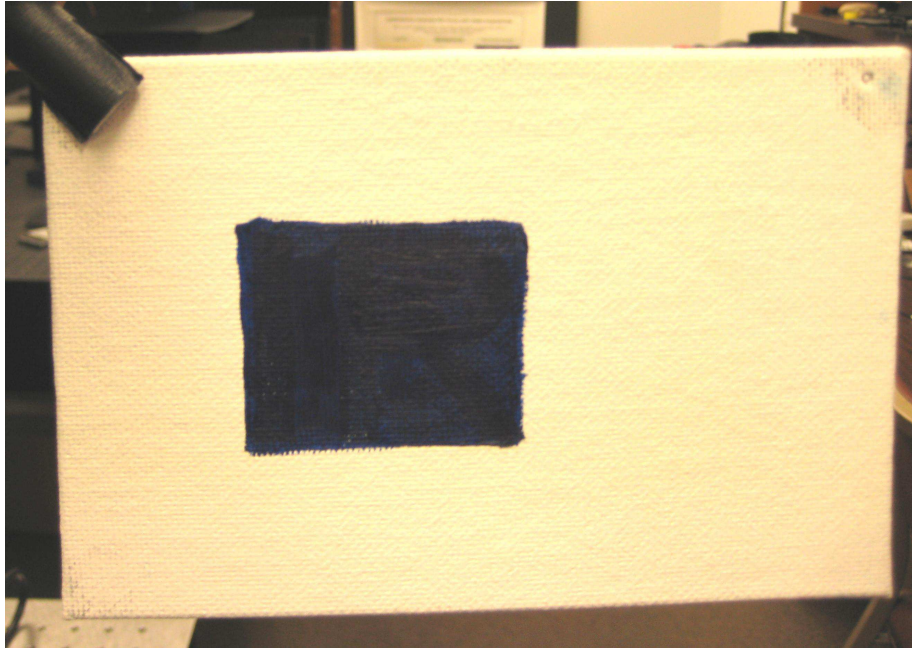
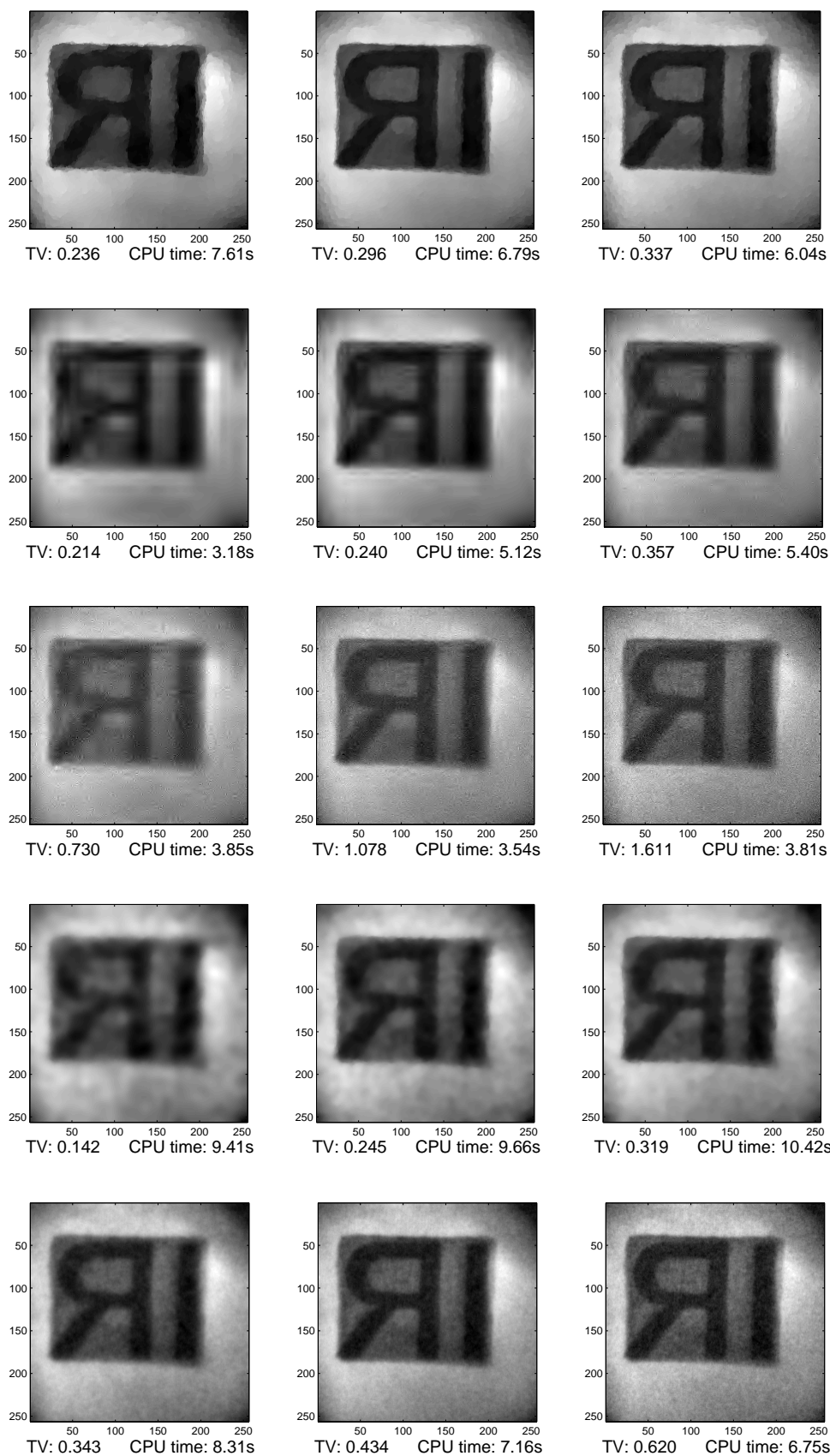


Figure 4.7: Real target in visible light.

ordered Walsh Hadamard transform. The data generated by the single pixel camera are more complicated and harder to reconstruct since various sources of noise are introduced which might be caused by environment, equipment, and so forth. Besides, noise level is also usually unpredictable. Therefore, most of theoretical tricks to estimate parameters based on the type and level of noise become helpless in practice.

As we mentioned before, one of the most significant advantages of the single pixel camera is to reduce the cost of infrared cameras. Test 7 (corresponding to Figure 4.8), demonstrates an infrared image recovery. A canvas board with the characters “IR” written on it by charcoal pencil was entirely covered by the blue oil paint which results in invisibility of “IR” to human eyes or to ordinary cameras as indicated in Figure 4.7. This board was illuminated by a 150 watt halogen lamp and picture was taken by the single pixel camera [44]. We respectively applied TVAL3, FPC_bb, YALL1, TwIST, NESTA, and ℓ_1 -Magic in sequence to 15%, 35%, and 50% data captured by the single



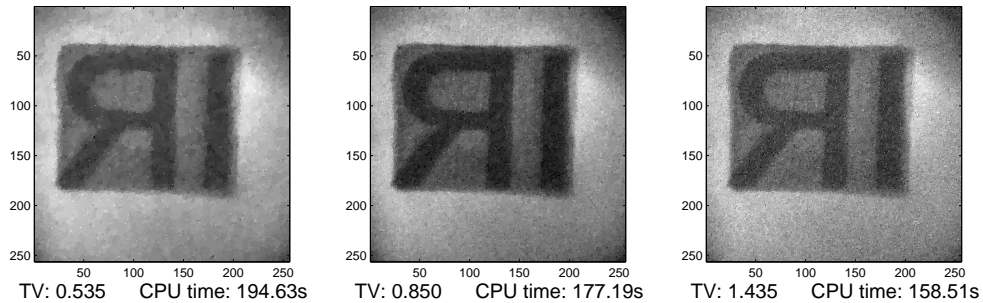


Figure 4.8: Recovered 256×256 infrared RI image. The six rows are reconstructed by TVAL3, FPC_bb, YALL1, TwIST, NESTA, and ℓ_1 -Magic respectively, for sampling ratios 15%, 35%, and 50%.

pixel camera to achieve approximate images.

The tolerance is uniformly fixed to $1e - 2$. All other parameters are set as default except the following ones: $opts.mu = 8$ and $opts.beta = 80$ for TVAL3; $opts.nu = .6$ for YALL1; $tau = 1/4000$ for TwIST; $mu = .02$, $Lambda = .01$, $La = 1$ for NESTA; $mu = 2$ for ℓ_1 -Magic.

Scrutinizing Figure 4.8, the following facts are observed: TV solvers can recover the edges better, make recovered images look sharper, and provide better contrast than ℓ_1 solvers in general. Among TV solvers, TwIST and ℓ_1 -Magic is inferior in this example since images recovered by TwIST are hard to recognize when measurement rate is low and ℓ_1 -Magic always requires at least 10 times longer running time than others. NESTA and TVAL3 are capable of successful reconstruction whatever the measurement rate is and require fairly close running time, but the image recovered by TVAL3 is much sharper and cleaner than the one recovered by NESTA at each measurement rate which indicates TVAL3 is superior to NESTA in denoising in the process of reconstruction. These facts manifest the power of TVAL3 on SPC data in some sense.

As a laser-based failure-analysis technique, the traditional OBIC scans a focused

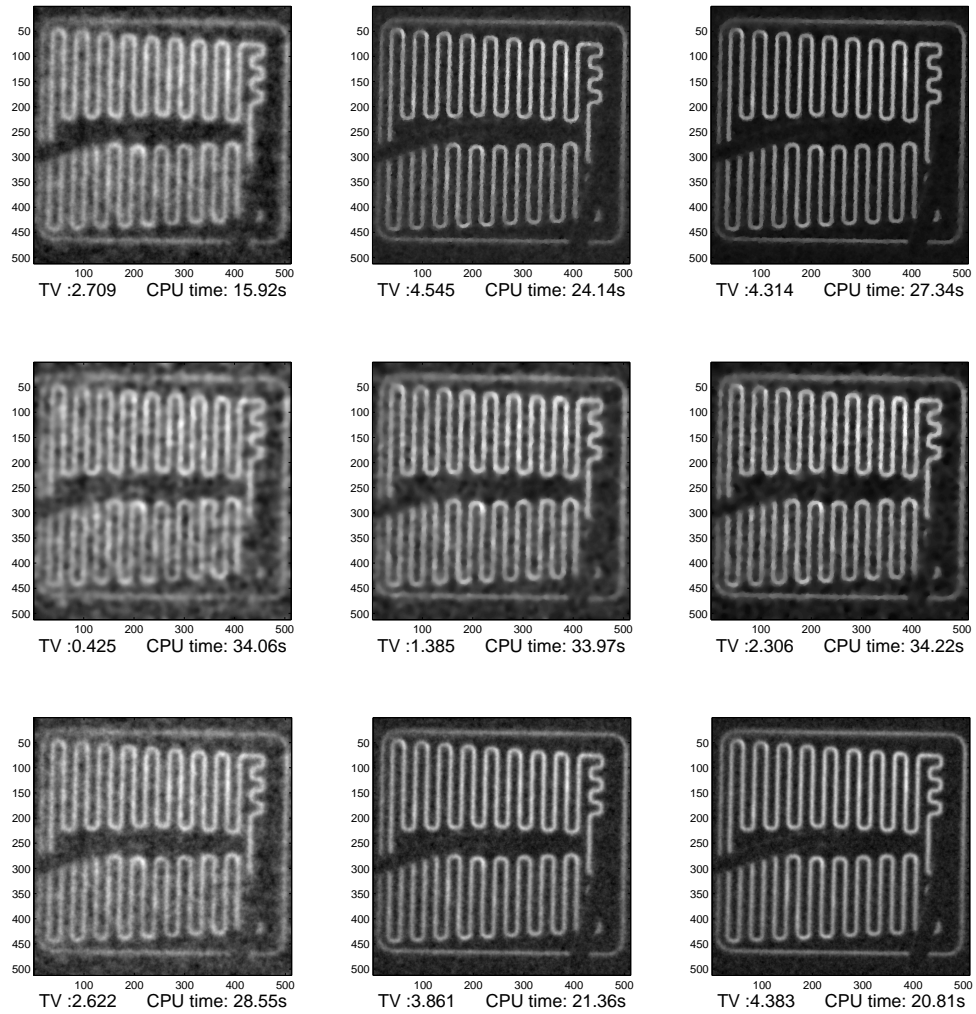


Figure 4.9: Recovered 512×512 discrete transistor image. The three rows are reconstructed by TVAL3, TwIST, and NESTA respectively, for sampling ratios 5%, 15%, and 24%.

laser beam across a sample by means of a laser scanning microscope (LSM). Inspired by the single pixel camera, we suggested a new compressive sensing method to acquire the same data with no need of a laser or an LSM in [45]. Test 8 (corresponding to Figure 4.9) demonstrates OBIC signal recovery, which is a key step for this compressive sensing method. The experiment was set up as follows: an arc lamp was collimated onto a DMD, and the the DMD was imaged onto a discrete transistor under test to create structured illumination matching the digital pattern of the DMD. The OBIC signal from the discrete transistor was recorded by an analog-to-digital converter and reconstructed by compressive sensing solvers. The measurements here which decide the pattern of the DMD are the same as being used in the single pixel camera. Since ℓ_1 -Magic is much slower, the other three TV solvers—TVAL3, TwIST, and NESTA—are applied to this OBIC signal in test 8.

We set $opts.mu = 16$ and $opts.beta = 8$ for TVAL3; $tau = 1/6000$ for TwIST; $mu = .002$, $Lambda = .001$, $La = 1$ for NESTA. Besides, we uniformly fix the tolerance to $5e - 3$ for all three solvers. Other parameters are chosen as default.

Figure 4.9 validates the fact that TVAL3 is preferable to TwIST and NESTA in virtue of better edge-preserving and denosing effects. TVAL3 and NESTA spent slightly shorter running time than TwIST in this test.

Test 7 and 8 illustrate the advantages of TVAL3 in efficiency and denoising effect in contrast to other TV and ℓ_1 solvers in a practical setting, and substantiate that TVAL3 should be adopted as the core reconstruction solver of the single pixel camera.

4.4 Initial Tests on Complex Signals and Nonnegativity Constraints

TVAL3 also implemented subroutines to recover complex signals and settle nonnegativity constraints according to the same TVAL3 scheme described in Chapter 2. Though theoretical guarantee is as yet unobtainable, the numerical experiments well indicate that it is capable of the image reconstruction containing nonnegativity constraints and complex signal recovery even when the measurements are complex.

The following two tests take the permuted Fourier matrix as the measurement matrix which is complex and able to be carried out by means of fast Fourier transform. The additive Gaussian noise is enforced according to (4.1). Since none of the other TV solvers can be directly applied to complex signals encoded by complex measurements, we only demonstrate the results achieved by TVAL3.

Test 9 (corresponding to Figure 4.10) concentrates on a 1D complex staircase signal whose length is 65536 and number of jumps is 163. It is encoded by permuted Fourier matrix with 5% Gaussian noise in both real and complex parts, and then recovered from 25% measurements.

In TVAL3, we set parameters as default except for $opts.mu = 2^4$, $opts.beta = 2^5$, and $opts.tol = 1e - 3$.

Figure 4.10 shows that both the real part and the complex part of the signal under test are fully recovered in only a few seconds, which substantiates the efficiency and the robustness of TVAL3 even for the complex case.

Test 10 (corresponding to Figure 4.11) demonstrates an experiment to recover a 512×512 thorax image scanned by CT. It is also encoded by permuted Fourier matrix imposing 15% Gaussian noise to generate a complex observation f . The CT thorax image is restored by TVAL3 from 10% measurements. Since each pixel of

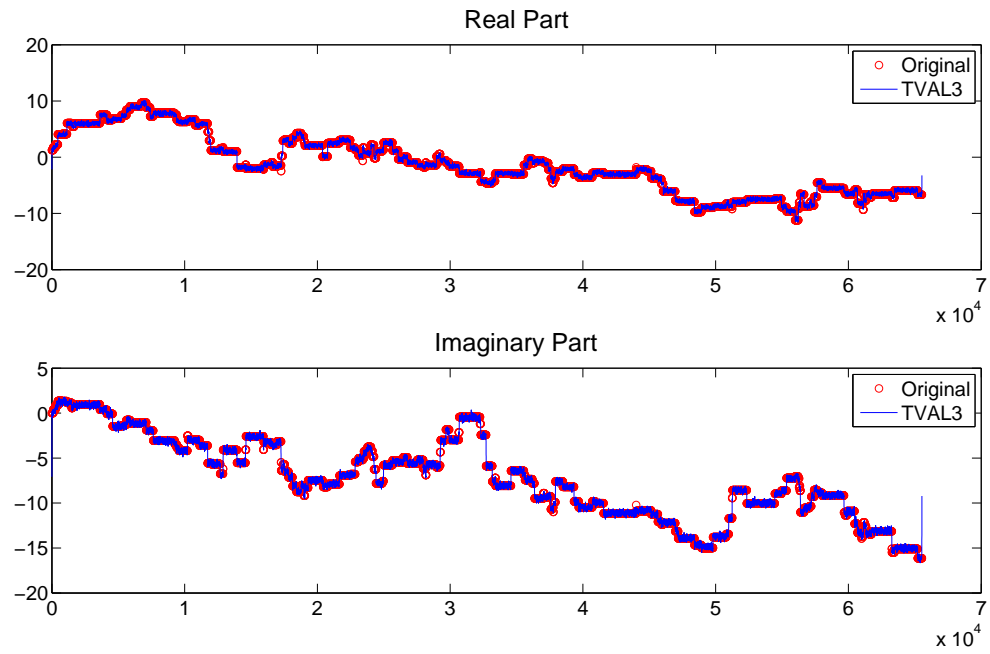


Figure 4.10: Recovered 1D complex staircase signal from 25% measurements. The noise level is 5%. Relative error recovered by TVAL3 is 2.92%, and running time is 8.70s.

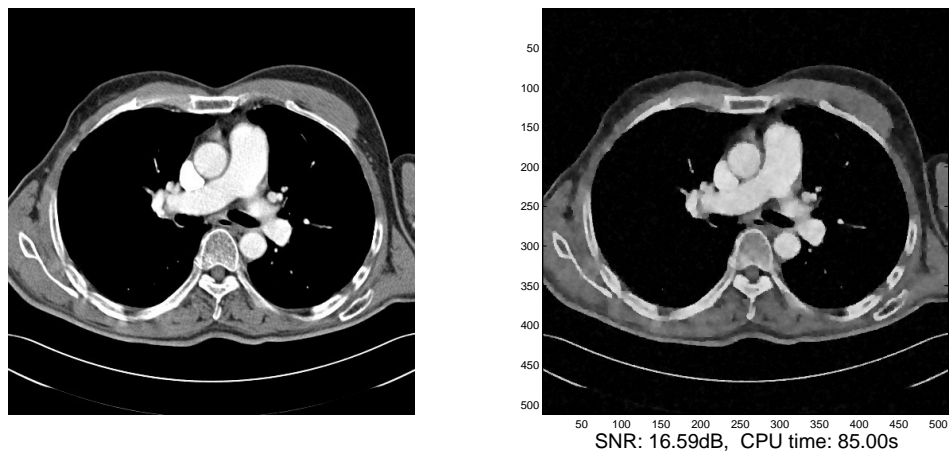


Figure 4.11: Recovered 512×512 CT thorax image from 10% measurements using TVAL3. The noise level is 15%.

this image is nonnegative, we can apply nonnegativity constraints on it and use the corresponding subroutine.

The parameter settings for TVAL3 are as follows: $opts.mu = 2^9$, $opts.beta = 2^7$, $opts.tol = 1e-4$, and $opts.nonneg = true$ to trigger the subroutine for nonnegativity cases. Others are assigned as default.

Examining Figure 4.11 carefully, we discover that most details of the CT thorax image has been restored only from 10% measurements. More precisely, every bright spot on the right side of the original image is still distinguishable on the recovered one. Furthermore, there are three very tiny bright spots on the left side of the original image, and one of them can still be visually recognized. These small details are extremely hard to recover when measurement rate is low or noise level is high, but might play a pivotal role for disease diagnosis.

These two tests numerically validate the convergence of extended algorithms to handle complex signals and nonnegativity constraints, respectively, although further investigation is required theoretically.

4.5 Discussions

TVAL3 scheme and its corresponding solver have been presented in detail and favorably compared with other state-of-the-art solvers. Its efficiency and robustness have been sufficiently substantiated by above experiments. Furthermore, TVAL3 scheme has exhibited its better denoising effects while reconstructing the measured data. Since the implementation of TVAL3 is considerably flexible, it can be used employing fast transforms, can solve many variants of the TV model, and even can reconstruct complex signals encoded by complex measurements. Due to its merits in efficiency, robustness, and denoising effects, TVAL3 is competent for the single pixel camera and

other related devices as the core reconstruction solver. Besides, TVAL3 is capable of medical image processing and other related compressive sensing applications.

How to choose optimal parameters without knowing the true solution and noise level has always been a big issue for almost every TV or ℓ_1 solvers. Fortunately, TVAL3 is not very sensitive to the fluctuation of parameters, which somehow reduces the difficulty to manipulate this solver for engineers and researchers. Research on this issue as well as the theoretical analysis of the algorithms is still in process.

Chapter 5

Future Work

The TVAL3 scheme has been stated in detail in Chapter 2. A large number of numerical experiments reported in Chapter 4 have shown their corresponding algorithms succeed in reconstructing images and surpassing other comparable algorithms in both running time and quality of recovered images. However, the theoretical analysis on convergence and convergence rate of the TVAL3 scheme has not yet been fully investigated. Thus, one of primary tasks in the next stage is to prove the convergence and discover the convergence rate of this scheme. More precisely, Local Convergence Theorem 1 indicates the convergence of Algorithm 3, as long as the convergence of the alternating minimization scheme mentioned in Algorithm 2 can be proven. This proposed work would complete the TVAL3 scheme and provide the theoretical guarantee for further extensions.

In the course of studying the TVAL3 scheme, there are two other related topics which have drawn my attention and might enrich my Ph.D. research. In particular, one is if it is possible to extend the TVAL3 scheme to 3D or hyperspectral image reconstruction; the other is how to develop a new algorithm solving the dual problem of a TV model with the aid of the TVAL3 scheme. These two issues will be proposed

in detail in the next two sections.

5.1 Hyperspectral Imaging

Over the past decade, more and more researchers dedicate themselves to the investigation of hyperspectral imaging. It has matured into one of the most powerful and fastest growing technologies. For example, the development of hyperspectral sensors and their corresponding software to analyze hyperspectral data has been regarded as a critical breakthrough in the field of remote sensing. However, it is usually intractable to collect and store hyperspectral data. I intend to explore if compressive sensing algorithms such as the TVAL3 scheme could be extended to help increase the efficiency of hyperspectral data collection and storage.

The basic concepts of hyperspectral imaging will be introduced in Section 5.1.1 and mathematical formulation will be derived in Section 5.1.2.

5.1.1 Basic Concepts

By exploiting the wavelength composition of electromagnetic radiation (EMR), hyperspectral imaging collects and processes data from across the electromagnetic spectrum. Hyperspectral sensors capture information as a series of “images”. Each image represents a spectral band which is a range of the electromagnetic spectrum. These images generated from different bands pile up and form a 3D hyperspectral cube for processing and further analysis. If each image can be viewed as a long vector, the hyperspectral cube will become a big matrix which is more easily accessible mathematically. Each column of the matrix records the information from the same spectral band and each row records the information at the same pixel. For much of the past decade, hyperspectral imaging has been an active research topic and widely devel-

oped. It has a lot of applications on industry, agriculture, and military, such as mineral exploration, food inspection, camouflage detection, environmental monitoring, pharmaceutical manufacturing, resource management, and so forth.

The fundamental property of hyperspectral imaging which researchers want to obtain is *spectral reflectance*: the ratio of reflected energy to incident energy as a function of wavelength [92]. Reflectance varies with wavelength for most materials. These variations are evident and sometimes characteristic while comparing spectral reflectance plots versus wavelength for different materials. Several libraries of reflectance spectra of natural and man-made materials are accessible for public use, such as ASTER Spectral Library [96] and USGS Spectral Library [97]. These libraries provide a source of reference spectra helping the interpretation and analysis of hyperspectral images.

However, it is highly possible that more than one material contributes to an individual spectrum captured by the sensor, which leads to a composite or mixed spectrum. The mixed spectrum can be decomposed into several *endmembers* which are defined as spectrally “pure” features, such as soil, vegetation, and so forth. In mineralogy, an *endmember* refers to a mineral at the extreme end of a mineral series in terms of purity. For example, albite ($\text{NaAlSi}_3\text{O}_8$) and anorthite ($\text{CaAl}_2\text{Si}_2\text{O}_8$) are two endmembers in the plagioclase series of minerals.

If the endmember spectra are available beforehand, we can mathematically decompose each pixel’s spectrum of a hyperspectral image to identify the relative abundance of each endmember component. This process is called “unmixing”. However, the challenge is how to identify a set of spectral endmembers that correspond to actual physical components. It becomes even harder to identify without the aid of prior information. Unmixing the hyperspectral image without aware of endmember spectral and even the number of endmembers is called “blind unmixing”. Linear unmixing is

a simple spectral matching approach, whose underlying premise is that a relatively small number of common endmembers are involved in a scene, and most spectral variability in this scene can be attributed to spatial mixing of these endmember components in distinct proportions.

Since the enormous volume of hyperspectral data, it is always hard to process and analyze in real time. Each image corresponding to some spectral band of hyperspectral data is compressible and is able to be reconstructed from a relatively small amount of measurements. In fact, the concept of the single pixel camera can be extended to the acquisition of compressed hyperspectral data. A straightforward way can be described as follows: collect the compressed hyperspectral data; recover the hyperspectral cube from the compressed data by compressive sensing techniques; detect endmembers by unmixing algorithms. However, due to the massive amount of data included in hyperspectral cube, it is usually too costly to recover the entire cube. Besides, the cube becomes unnecessary once we have successfully detected endmembers. Can we decide endmembers directly from the compressed data without recovering the hyperspectral cube? Can we do it in an efficient manner? Can we utilize the spectral information to further compress the data? All these open questions are challenging and require long-term research. First, the problem is formulated in optimization in the next section.

5.1.2 Initial Formulation

Suppose that $X \in \mathbb{R}^{n_p \times n_b}$ is a unknown matrix representing n_p -pixel by n_b -band, hyperspectral image cube, $F \in \mathbb{R}^{m \times n_b}$ represents the observation data, $A \in \mathbb{R}^{m \times n_p}$ is a measurement matrix with $m < n_p$, and $\Omega \in \mathbb{R}^{m \times n_b}$ is random noise. Then they are

combined by the following data acquisition model:

$$F = AX + \Omega.$$

To proceed blind unmixing, it is necessary to assume that the image cube X has a low-dimensional representation

$$X = HW, \quad H, W \geq 0, \quad H\mathbf{1} = \mathbf{1},$$

where $H \in \mathbb{R}_+^{n_p \times n_e}$, $W \in \mathbb{R}_+^{n_e \times n_b}$, and $\mathbf{1}$ is the vector of all ones. Here n_e is an estimated number of endmembers that should be far less than both n_p and n_b . Each row of W represents an endmember spectrum, and each row of H consists of abundance coefficients for a pixel.

According to the theory of compressive sensing, H and W might be recovered from the following optimization model:

$$\begin{aligned} \min_{H, W} \quad & R_{hw}(HW) + R_h(H) + R_w(W) + \frac{1}{2} \|AHW - F\|_F^2 \\ \text{s.t.} \quad & H, W \geq 0, \quad H\mathbf{1} = \mathbf{1}, \end{aligned} \quad (5.1)$$

where $R_{hw}(\cdot)$ is a joint regularization function for the product HW , $R_h(\cdot)$ and $R_w(\cdot)$ are individual regularization functions for H and W , respectively.

How to appropriately choose three regularization functions might be as essential as designing the algorithm. Opportune regularization functions can not only guarantee the good recoverability, but also help discover an efficient algorithm. Proposing the following regularization functions in a TV manner may be appropriate and worthy to

further investigate:

$$R_{hw}(HW) = \alpha_{hw} \sum_{j \in J \subset \{1, 2, \dots, n_b\}} \text{TV}(HW e_j)$$

$$R_h(H) = \alpha_h \sum_{j=1}^{n_e} \text{TV}(H e_j)$$

$$R_w(W) = \alpha_w \sum_{j=1}^{n_e} \|\Psi_w W^T e_j\|_1$$

where $\Psi_w \in \mathbb{R}^{n_b \times n_b}$ is an appropriate sparsifying basis, and α 's are nonnegative balancing parameters. More precisely, one way to sparsify W is to introduce the second-order total variation in 1D

$$\text{TV}_2(u) = \sum_{i=1}^n |\Delta_i^2 u|,$$

where $\Delta_i^2 u = u_{i+2} - 2u_{i+1} + u_i$ is the second-order derivative approximation of u at i th position for $u \in \mathbb{R}^n$. Then we can define

$$R_w(W) = \alpha_w \sum_{j=1}^{n_e} \text{TV}_2(W^T e_j).$$

Since the minimization problem (5.1) is bi-convex, it is amenable to alternating minimization, i.e., minimizing with respect to W while fixing H and vice versa. The two subproblems are

$$\min_H R_{hw}(HW) + R_h(H) + \frac{\mu}{2} \|AHW - F\|_F^2 \quad \text{s.t.} \quad H \geq 0, \quad H\mathbf{1} = \mathbf{1}, \quad (5.2)$$

$$\min_W R_{hw}(HW) + R_w(W) + \frac{\mu}{2} \|AHW - F\|_F^2 \quad \text{s.t.} \quad W \geq 0. \quad (5.3)$$

It appears that the two subproblems are both convex but not further separable in terms of their rows or columns. In general, $n_p \gg n_b$ and $H \in \mathbb{R}_+^{n_p \times n_e}$ is the larger variable, consisting of n_e images of n_p pixels. An algorithm which can efficiently solve this problem requires further study.

Successfully solving this problem could not only make a big process in hyperspectral imaging, but also inspire the innovation in the other 3D data processing.

5.1.3 Parallel Algorithms and Implementations on High Performance Computers

In recent years, because of the advances in sensor technology, hyperspectral imaging has been further developed and is able to collect hundreds of images corresponding to different wavelength channels. With the aid of such detailed spectral information, the ability in detection and identification of materials will be significantly improved. However, the massive amount of data prohibits efficient storage and even other operations. For example, compared to the regular image reconstruction, the complexity of each operation on hyperspectral data increases by n_b times, where n_b represents the number of channels as mentioned before. Suppose that data is collected under 180 different channels and the unmixing algorithm is as efficient as the reconstruction algorithm such as TVAL3. Then it may take half an hour to unmix a hyperspectral image while taking only 10 seconds to reconstruct a regular image with the same resolution. The storage of these massive data would be another issue.

Parallel computing refers to the simultaneous use of multiple compute resources to solve a computational problem. It requires a single computer with multiple processors or multiple computers connected by a network. Specifically, a computational problem is divided into discrete parts and each part is further broken down to a series of

instructions which from each part can execute simultaneously on different processors or computers. Implementation of parallel computing would shorten the running time significantly.

Since the massive amount of data is involved in the minimization problem (5.1), it provides the potential to develop a parallel unmixing algorithm. Before that, the subproblems (5.2) and (5.3) need to be investigated and solved in an efficient way in order to propose a serial unmixing algorithm. The way of implementing parallel computing correlates to the structure of the serial algorithm. Some existing parallelizing techniques (see [93], for example) also improve opportunities for exploiting high-performance parallel algorithms.

The Research Computing Support Group (RCSG) at Rice University provides shared computing services including Ada, SUG@R, and STIC. Taking advantages of the RCSG resources would greatly help the design and test of parallel algorithms, which will be an important subject in my future research.

5.2 Exploration on Dual Method

To study the dual problem of a TV model, let us first restate the TV model (2.1) for compressive sensing in the complex domain:

$$\min_u \sum_i \|D_i u\|_p, \quad \text{s.t. } Au = b, \quad (5.4)$$

where $p \in \mathcal{R}$, $p \geq 1$, $u \in \mathcal{C}^n$ or $u \in \mathcal{C}^{s \times t}$ with $s \cdot t = n$, and $A \in \mathcal{C}^{m \times n}$.

The dual problem of this TV model will be derived in Section 5.2.1 and the initial method to the dual problem will be suggested in Section 5.2.2.

5.2.1 Derivation of Dual Problem

Suppose that $q \in \mathcal{R}^+$ satisfies

$$\frac{1}{p} + \frac{1}{q} = 1.$$

According to the Hölder's Inequality,

$$\|x\|_p = \max_{\|y\|_q \leq 1} | \langle y, x \rangle |, \quad (5.5)$$

where $x, y \in \mathcal{C}^N$.

As we all known, for any $x, y \in \mathcal{C}^N$,

$$| \langle y, x \rangle | \geq \mathbf{Re}(\langle y, x \rangle).$$

where \mathbf{Re} represents the real part operator. Thus,

$$\|x\|_p = \max_{\|y\|_q \leq 1} | \langle y, x \rangle | \geq \max_{\|y\|_q \leq 1} \mathbf{Re}(\langle y, x \rangle). \quad (5.6)$$

The maximizer of (5.5) \tilde{y} will be achieved while $\tilde{y}_k = cx_k|x_k|^{p-2}$. Choose constant c such that $\|\tilde{y}\|_q = 1$. Under these circumstances,

$$\|x\|_p = \langle \tilde{y}, x \rangle = \mathbf{Re}(\langle \tilde{y}, x \rangle). \quad (5.7)$$

In the light of (5.6) and (5.7),

$$\|x\|_p = \max_{\|y\|_q \leq 1} \mathbf{Re}(\langle y, x \rangle).$$

Based on the above fact,

$$\begin{aligned}
\sum_{i=1}^n \|D_i u\|_p &= \sum_{i=1}^n \max_{\|v_i\|_q \leq 1} \mathbf{Re}(\langle v_i, D_i u \rangle) \\
&= \max_{\|v_i\|_q \leq 1} \sum_{i=1}^n \mathbf{Re}(\langle v_i, D_i u \rangle) \\
&= \max_{\|v_i\|_q \leq 1} \sum_{i=1}^n \mathbf{Re}(\langle D_i^* v_i, u \rangle).
\end{aligned}$$

As a matter of fact, $\mathbf{Re}(\langle D_i^* v_i, u \rangle)$ is bilinear for any i , which leads to,

$$\begin{aligned}
\max_{\|v_i\|_q \leq 1} \sum_{i=1}^n \mathbf{Re}(\langle D_i^* v_i, u \rangle) &= \max_{\|v_i\|_q \leq 1} \mathbf{Re}\left(\sum_{i=1}^n \langle D_i^* v_i, u \rangle\right) \\
&= \max_{\|v_i\|_q \leq 1} \mathbf{Re}\left(\left\langle \sum_{i=1}^n D_i^* v_i, u \right\rangle\right).
\end{aligned}$$

Therefore,

$$\min_{Au=f} \sum_{i=1}^n \|D_i u\|_p \iff \min_{Au=f} \max_{\|v_i\|_q \leq 1} \mathbf{Re}\left(\left\langle \sum_{i=1}^n D_i^* v_i, u \right\rangle\right). \quad (5.8)$$

In 1958, Sion generalized distinguished John Von Neumann's minimax theorem [87] in the theory of simultaneous games as following:

Theorem 4 (Sion's Minimax Theorem [88]). *Let X be a compact convex subset of a linear topological space and Y a convex subset of a linear topological space. If f is a real-valued function on $X \times Y$ with the property that $f(x, \cdot)$ is upper semicontinuous and quasiconcave on Y , $\forall x \in X$, and $f(\cdot, y)$ is lower semicontinuous and quasi-convex on X , $\forall y \in Y$, then,*

$$\min_{x \in X} \max_{y \in Y} f(x, y) = \max_{y \in Y} \min_{x \in X} f(x, y).$$

Proof. See [88, 89, 90, 91] for different proofs. □

A straightforward analysis indicates that

- $\{u : Au = f\}$ is convex,
- $\{v = \{v_1, v_2, \dots, v_n\} : \|v_i\|_q \leq 1 \ \forall 1 \leq i \leq n\}$ is compact convex,
- and $f(v, u) = \mathbf{Re}(\langle \sum_{i=1}^n D_i^* v_i, u \rangle)$ is bilinear.

These three facts suggest that the Sion's minimax theorem can be used to exchange min and max of (5.8); i.e.,

$$\begin{aligned} \min_{Au=f} \sum_{i=1}^n \|D_i u\|_p &\iff \min_{Au=f} \max_{\|v_i\|_q \leq 1} \mathbf{Re}(\langle \sum_{i=1}^n D_i^* v_i, u \rangle) \\ &\iff \max_{\|v_i\|_q \leq 1} \min_{Au=f} \mathbf{Re}(\langle \sum_{i=1}^n D_i^* v_i, u \rangle). \end{aligned}$$

For the inner minimization, if there exists $z \in \mathcal{C}^m$, s.t.

$$A^* z = \sum_{i=1}^n D_i^* v_i,$$

then

$$\begin{aligned} \min_{Au=f} \mathbf{Re}(\langle \sum_{i=1}^n D_i^* v_i, u \rangle) &= \min_{Au=f} \mathbf{Re}(\langle A^* z, u \rangle) \\ &= \min_{Au=f} \mathbf{Re}(\langle z, Au \rangle) \\ &= \mathbf{Re}(\langle z, f \rangle). \end{aligned}$$

Otherwise,

$$\min_{Au=f} \mathbf{Re}(\langle \sum_{i=1}^n D_i^* v_i, u \rangle) = -\infty.$$

Therefore, the dual problem of the TV model is

$$\max_{v_i, z} \mathbf{Re}(\langle z, f \rangle), \quad \text{s.t. } \|v_i\|_q \leq 1 \text{ and } A^*z = \sum_{i=1}^n D_i^* v_i. \quad (5.9)$$

Furthermore, the whole derivation is still correct if D_i s for all i are some general linear operators in $\mathcal{C}^{r \times n}$. The conclusion on the dual problem can be easily extended to the model with general linear operators, without modifying anything.

Since the primal problem (5.4) is convex and there always exists at least one strictly feasible point for (5.4), the optimal duality gap between the primal problem (5.4) and the dual problem (5.9) is zero, i.e., the strong duality holds.

5.2.2 Methodology on Dual Problem

Restricted to the real domain, the dual problem (5.9) can be rewritten as

$$\min_{v_i, z} -f^T z, \quad \text{s.t. } \|v_i\|_q \leq 1 \text{ and } A^T z = \sum_i D_i^T v_i. \quad (5.10)$$

The augmented Lagrangian method has been well studied in Section 2.1. This method requires minimizing the corresponding augmented Lagrangian function at each iteration. Therefore, the associated subproblem of solving (5.10) by the augmented Lagrangian method is

$$\begin{aligned} \min_{v_i, z} \mathcal{L}_D(v_i, z) &\triangleq -f^T z - \omega^T (A^T z - \sum_i D_i^T v_i) \\ &+ \frac{\gamma}{2} \|A^T z - \sum_i D_i^T v_i\|_2^2, \quad \text{s.t. } \|v_i\|_q \leq 1. \end{aligned} \quad (5.11)$$

If this subproblem can be solved efficiently, it is highly likely to render a new creditable algorithm for the dual problem (5.10).

Mirroring the TVAL3 scheme, we can try to apply the alternating direction method to (5.11). That means two subproblems need to be settled alternately:

$$\min_z \mathcal{L}_D(v_i, z),$$

and

$$\min_{v_i} \mathcal{L}_D(v_i, z), \quad \text{s.t. } \|v_i\|_q \leq 1.$$

After simplification, they respectively correspond to

$$\min_z -(f + A\omega)^T z + \frac{\gamma}{2} \|A^T z - \sum_i D_i^T v_i\|_2^2, \quad (5.12)$$

and

$$\min_{v_i} \sum_i (D_i \omega)^T v_i + \frac{\gamma}{2} \|A^T z - \sum_i D_i^T v_i\|_2^2, \quad \text{s.t. } \|v_i\|_q \leq 1, \quad (5.13)$$

For (5.12), the one-step steepest descent scheme proposed in Section 2.3.2 should work to obtain a roughly approximate minimizer. However, further investigation is needful in the future to solve (5.13) exactly or approximately.

In fact, it is likely to incorporate other methods or algorithms to settle (5.11) or even (5.10) properly, which also demand further research in times to come.

Bibliography

- [1] E. Candès, and T. Tao, *Decoding by linear programming*, IEEE Trans. Inform. Theory, vol. 51, no. 12, pp. 4203–4215, 2005.
- [2] E. Candès, and T. Tao, *Near optimal signal recovery from random projections: Universal encoding strategies*, IEEE Trans. on Inform. Theory, vol. 52, no. 12, pp. 5406–5425, 2006.
- [3] E. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, vol. 52, no. 2, pp. 489–509, 2006.
- [4] D. Donoho, *Compressed sensing*, IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, 2006.
- [5] Y. Zhang, *On theory of compressive sensing via ℓ_1 -minimization: Simple derivations and extensions*, CAAM Technical Report TR08-11, Department of Computational and Applied Mathematics, Rice University, July 2008.
- [6] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, pp. 259–268, 1992.
- [7] F. Santosa and W. W. Symes, *Linear inversion of band-limited reflection seismograms*, SIAM J. Sci. Statist. Comput., vol. 7, no. 4, pp. 1307–1330, 1986.
- [8] D. L. Donoho and P. B. Stark, *Uncertainty principles and signal recovery*, SIAM J. Appl. Math., vol. 49, pp. 906–931, 1989.
- [9] D. L. Donoho and B. F. Logan, *Signal recovery and the large sieve*, SIAM J. Appl. Math., vol. 52, pp. 577–591, 1992.
- [10] D. L. Donoho and X. Huo, *Uncertainty principles and ideal atomic decomposition*, IEEE Trans. Inform. Theory, vol. 47, pp. 2845–2862, 2001.
- [11] D. L. Donoho and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization*, Proc. Natl. Acad. Sci. USA, vol. 100, pp. 2197–2202, 2003.

- [12] M. Elad and A. M. Bruckstein, *A generalized uncertainty principle and sparse representation in pairs of \mathbb{R}^N bases*, IEEE Trans. Inform. Theory, vol. 48, 2558–2567, 2002.
- [13] A. Feuer and A. Nemirovski, *On sparse representation in pairs of bases*, IEEE Trans. Inform. Theory, vol. 49, 1579–1581, 2003.
- [14] R. Gribonval and M. Nielsen, *Sparse representations in unions of bases*, IEEE Trans. Inform. Theory, vol. 49, 3320–3325, 2003.
- [15] J. J. Fuchs, *On sparse representations in arbitrary redundant bases*, IEEE Trans. Inform. Theory, vol. 50, 1341–1344, 2004.
- [16] Yu. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., Ser. A 103, pp. 127–152, 2005.
- [17] E. Hale, W. Yin, and Y. Zhang, *Fixed-point continuation for $L1$ -minimization: Methodology and convergence*, SIAM J. Optim., vol. 19, no. 3, 2008.
- [18] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imag. Sci., vol. 1, no. 4, pp. 248–272, 2008.
- [19] 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.
- [20] R. Courant, *Variational methods for the solution of problems with equilibrium and vibration*, Bull. Amer. Math. Soc., vol. 49, pp. 1–23, 1943.
- [21] M. Rudelson and R. Vershynin, *Geometric approach to error-correcting codes and reconstruction of signals*, Int. Math. Res. Not., vol. 64, pp. 4019–4041, 2005.
- [22] D. Donoho and J. Tanner, *Neighborliness of randomly-projected simplices in high dimensions*, Proc. National Academy of Sciences, vol. 102, no. 27, pp. 9452–9457, 2005.
- [23] E. J. Candès, M. B. Wakin, and S. Boyd, *Enhancing sparsity by reweighted ell_1 minimization*, Journal of Fourier Analysis and Applications, vol. 14, no. 5, pp. 877–905, December 2008.
- [24] A. Chambolle, *An algorithm for total variation minimization and applications*, Journal of Mathematical Imaging and Vision, vol. 20, 89–97, Jan. 2004.
- [25] A. Chambolle and P. L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., vol. 76, pp. 167–188, 1997.

- [26] T. F. Chan, S. Esedoglu, F. Park, and A. Yip, *Recent developments in total variation image restoration*, CAM Report 05-01, Department of Mathematics, UCLA, 2004.
- [27] J. Tropp, *Just relax: Convex programming methods for identifying sparse signals*, IEEE Transactions on Information Theory, vol. 51, pp. 1030–1051, 2006.
- [28] Y. Tsaig and D. Donoho, *Extensions of compressed sensing*, Signal Processing, vol. 86, no. 3, pp. 533–548, 2005.
- [29] M. Lustig, D. Donoho, J. Pauly, *Sparse MRI: The application of compressed sensing for rapid MR imaging* Magnetic Resonance in Medicine, vol. 58 no. 6 pp. 1182–1195, 2007.
- [30] T. Chang, L. He, and T. Fang, *MR image reconstruction from sparse radial samples using bregman iteration*, ISMRM, 2006.
- [31] J. Laska, S. Kirolos, M. Duarte, T. Ragheb, R. Baraniuk, and Y. Massoud, *Theory and implementaion of an analog-to-information converter using random demodulation*, In Proceedings of the IEEE International Symposium on Circuites and Systems (ISCAS), New Orleans, Louisiana, 2007.
- [32] D. Geman and G. Reynolds, *Constrained restoration and the recovery of discontinuities*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 3, pp. 367–383, 1992.
- [33] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization*, IEEE Transactions on Image Processing, vol. 4, no. 7, pp. 932–946, 1995.
- [34] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, *Distributed compressed sensing of jointly sparse signals*, in 39th Asilomar Conference on Signals, Systems and Computers, pp. 1537–1541, 2005.
- [35] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, *Compressed Sensing for Networked Data*, IEEE Signal Processing, vol. 25, no. 2, pp. 92–101, March 2008.
- [36] J. Yang, Y. Zhang, and W. Yin, *A fast TVL1-L2 minimization algorithm for signal reconstruction from partial fourier data*, Tech. Report 08-27, CAAM, Rice University. Submitted to J-STSP.
- [37] J. Yang, W. Yin, Y. Zhang, and Y. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, Tech. Report 08-09, CAAM, Rice University, Submitted to SIIMS.

- [38] J. Yang, Y. Zhang, and W. Yin, *An efficient TVL1 algorithm for deblurring of multichannel images corrupted by impulsive noise*, TR08-12, CAAM, Rice University, Submitted to SISC.
- [39] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk, *A new compressive imaging camera architecture using optical-domain compression*, Computational Imaging IV, vol. 6065, pp. 43–52, Jan. 2006.
- [40] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, *An architecture for compressive imaging*, 2006 IEEE International Conference on Image Processing, pp. 1273–1276, Oct. 2006.
- [41] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, *Compressive imaging for video representation and coding*, Picture Coding Symposium–PCS 2006, Beijing, China, April 2006.
- [42] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, *Single pixel imaging via compressive sampling*, IEEE Signal Processing, vol. 25, no. 2, pp. 83–91, 2008.
- [43] J. Sampsell, *An overview of the digital micromirror device (DMD) and its application to projection displays*, 1993 SID International Symposium Digest of Technical Papers, vol. 24, pp. 1012, 1993.
- [44] T. Sun, C. Li, Y. Zhang, and K. F. Kelly, *Infrared Imaging by a Single Photodiode based on Compressive Sensing*, submitted to Applied Physics Letters, 2009.
- [45] T. Sun, G. L. Woods, C. Li, Y. Zhang, and K. F. Kelly, *OBIC Measurements without Lasers or Raster-Scanning Based on Compressive Sensing*, accepted by International Symposium on Testing and Failure Analysis (ISTFA), September 2009.
- [46] B. K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM Journal on Computing, vol. 24, pp. 227–234, 1995.
- [47] G. S. Robinson, *Logical convolution and discrete Walsh and Fourierpower spectra*, IEEE Trans. Audio Electroacoust., vol. AU–20, pp. 271–280, Oct. 1972.
- [48] F. Gray, *Pulse code communication*, U.S. Patent 2632058, March 17, 1953 (filed Nov. 1947).
- [49] K. G. Beauchamp, *Applications of Walsh and Related Functions*, Academic Press Inc., 1984.

- [50] P. E. Black, "greedy algorithm" in *Dictionary of Algorithms and Data Structures*[online], U.S. National Institute of Standards and Technology, February 2005.
- [51] S. G. Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries*, IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3397–3415, 1993.
- [52] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*, In Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, 1993.
- [53] S. S. Chen, *Basis Pursuit*, PhD thesis, Stanford University, Department of Statistics, 1995.
- [54] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Review, vol. 43, no. 1, pp. 129–159, 2001.
- [55] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [56] S. Becker, J. Bobin, and E. Candès, *NESTA: A Fast and Accurate First-order Method for Sparse Recovery*, Technical Report, California Institute of Technology, April 2009.
- [57] J. Bioucas-Dias, and M. Figueiredo, *A new TwIST: Two-step iterative thresholding algorithm for image restoration*, IEEE Trans. Imag. Process., vol. 16, no. 12, pp. 2992–3004, 2007.
- [58] J. Bioucas-Dias and M. Figueiredo, *Two-step algorithms for linear inverse problems with non-quadratic regularization*, IEEE International Conference on Image Processing–ICIP2007, San Antonio, TX, USA, September 2007.
- [59] Y. Zhang, *User's Guide for YALL1: Your ALgorithms for L1 Optimization*, Technical Report TR09-17, Department of Computational and Applied Mathematics, Rice University, May 2009.
- [60] R. Nowak and M. Figueiredo, *Fast wavelet-based image deconvolution using the EM algorithm*, in Proc. 35th Asilomar Conf. Signals, Systems, Computers, vol. 1, pp. 371–375, 2001.
- [61] J. Starck, M. Nguyen, and F. Murtagh, *Wavelets and curvelets for image deconvolution: A combined approach*, Signal Process., vol. 83, pp. 2279–2283, 2003.
- [62] I. Daubechies, M. Defriese, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Commun. Pure Appl. Math., vol. LVII, pp. 1413–1457, 2004.

- [63] P. Combettes and V. Wajs, *Signal recovery by proximal forward-backward splitting*, SIAM J. Multiscale Model. Simul., vol. 4, pp. 1168–1200, 2005.
- [64] M. Elad, *Why simple shrinkage is still relevant for redundant representations?*, IEEE Trans. Inf. Theory, vol. 52, no. 12, pp. 5559–5569, Dec. 2006.
- [65] V. N. Temlyakov, *Greedy algorithms and m -term approximation with regard to redundant dictionaries*, Journal of Approximation Theory, vol. 98, no. 1, pp. 117–145, 1999.
- [66] R. Gribonval and P. Vandergheynst, *On the exponential convergence of matching pursuits in quasi-incoherent dictionaries*, Technical Report 1619, IRISA, 2004.
- [67] J. A. Tropp, *Greedy is good: Algorithmic results for sparse approximation*, IEEE Transactions on Information Theory, vol. 50, no. 10, 2004.
- [68] W. Chan, M. Moravec, R. Baraniuk, and D. Mittleman, *Compressive sensing applied to homeland security*, IEEE Sensors Applications Symposium, Atlanta, GA, USA, February 2008.
- [69] D. W. Peaceman, H. H. Rachford, *The numerical solution of parabolic and elliptic differential equations*, Journal of the Society for Industrial and Applied Mathematics, vol. 3, pp. 28–41, 1955.
- [70] M. R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization Theory and Applications, vol. 4, pp. 303–320, and in Computing Methods in Optimization Problems, 2 (Eds L.A. Zadeh, L.W. Neustadt, and A.V. Balakrishnan), Academic Press, New York, 1969.
- [71] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, Optimization (Ed. R. Fletcher), Academic Press, London, New York, pp. 283–298, 1969.
- [72] R. T. Rockafellar, *The multiplier method of Hestenes and Powell applied to convex programming*, Journal of Optimization Theory and Applications, vol. 12, no. 6, pp. 555–562, 1973.
- [73] J. D. Buys, *Dual Algorithms for Constrained Optimization*, Rijksuniversiteit de Leiden, The Netherlands, PhD Thesis, 1972.
- [74] R. A. Tapia, *Newton's Method for Problems with Equality Constraints*, SIAM Journal on Numerical Analysis, vol. 11, pp. 174–196, 1974.
- [75] R. A. Tapia, *Newton's Method for Optimization Problems with Equality Constraints*, SIAM Journal on Numerical Analysis, vol. 11, pp. 874–886, 1974.

- [76] R. A. Tapia, *Diagonalized Multiplier Methods and Quasi-Newton Methods for Constrained Optimization*, Journal of Optimization Theory and Applications, vol. 22, no. 2, pp. 135–194, 1977.
- [77] R. H. Byrd, *Local Convergence of the Diagonalized Method of Multipliers*, Journal of Optimization Theory and Applications, vol. 26, no. 4, pp. 483–498, 1978.
- [78] S. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw-Hill Higher Education, Burr Ridge, IL, U.S.A., 1996.
- [79] D. Bertsekas, *Constraint Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.
- [80] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Mass., 1995.
- [81] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.
- [82] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal., vol. 8, pp. 141–148, 1988.
- [83] M. Raydan, *Convergence Properties of the Barzilai and Borwein Gradient Method*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, U.S.A., 1991.
- [84] H. Zhang and W. W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM J. Optim., vol. 14, pp. 1043–1056, 2004.
- [85] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., vol. 23, pp. 707–716, 1986.
- [86] P. H. Calamai and J. J. More, *Projected gradient methods for linearly constrained problems*, Math. Programming, vol. 39, pp. 93–116, 1987.
- [87] J. von Neumann, *Zur Theorie der Gesellschaftsspiele*, Math. Ann. vol. 100, pp. 295–320, 1928.
- [88] M. Sion, *On general minimax theorems*, Pac. J. Math., vol. 8, pp. 171–176, 1958.
- [89] K. Fan, *Sur un théorème minimax*, C.R. Acad. Sci., Paris, vol. 259, pp. 3925–3928, 1964.
- [90] W. Takahashi, *Nonlinear variational inequalities and fixed point theorems*, J. Math. Soc., Japan, vol. 28, pp. 168–181, 1976.
- [91] H. Komiya, *Elementary proof for Sion's minimax theorem*, Kodai Math. Journal, vol. 11, no. 1, pp. 5–7, 1988.

- [92] R. B. Smith, *Introduction to Hyperspectral Imaging*, MicroImages, Inc., July 2006.
- [93] G. Lou and S.-P. Han, *A parallel projection method for solving generalized linear least-squares problems*, *Numerische Mathematik*, vol. 53, no. 3, pp. 255–264, 1988.
- [94] <http://fourier.eng.hmc.edu/e161/lectures/wht/wht.html>.
- [95] http://en.wikipedia.org/wiki/Main_Page.
- [96] <http://speclib.jpl.nasa.gov>.
- [97] <http://speclab.cr.usgs.gov/spectral.lib04/spectral-lib04.html>.
- [98] <http://www.caam.rice.edu/~optimization/L1/TVAL3/>.