A Sparsity Enforcing Framework with TVL1 Regularization and its Application in MR Imaging and Source Localization

by

Wei Shen

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2011 by the
Graduate Supervisory Committee:

Hans D. Mittelmann, Chair
Anne Gelb
Rosemary Anne Renaut
Zdzislaw Jackiewicz
Christian Ringhofer

ARIZONA STATE UNIVERSITY

May 2011

ABSTRACT

The theme for this work is the development of fast numerical algorithms for sparse optimization as well as their applications in medical imaging and source localization using sensor array processing. Due to the recently proposed theory of Compressive Sensing (CS), the ℓ_1 minimization problem attracts more attention for its ability to exploit sparsity. Traditional interior point methods encounter difficulties in computation for solving the CS applications. In the first part of this work, a fast algorithm based on the augmented Lagrangian method for solving the large-scale TV-ℓ_1 regularized inverse problem is proposed. Specifically, by taking advantage of the separable structure, the original problem can be approximated via the sum of a series of simple functions with closed form solutions. A preconditioner for solving the block Toeplitz with Toeplitz block (BTTB) linear system is proposed to accelerate the computation. An in-depth discussion on the rate of convergence and the optimal parameter selection criteria is given. Numerical experiments are used to test the performance and the robustness of the proposed algorithm to a wide range of parameter values. Applications of the algorithm in magnetic resonance (MR) imaging and a comparison with other existing methods are included. The second part of this work is the application of the TV-ℓ_1 model in source localization using sensor arrays. The array output is reformulated into a sparse waveform via an over-complete basis and study the ℓ_p-norm properties in detecting the sparsity. An algorithm is proposed for minimizing a non-convex problem. According to the results of numerical experiments, the proposed algorithm with the aid of the ℓ_p-norm can resolve closely distributed sources with higher accuracy than other existing methods.

To my parents

ACKNOWLEDGEMENTS

I am especially grateful to my thesis advisor Professor Hans D. Mittelmann for directing me to do research in numerical optimization. Through the meetings and discussions, I have gained a lot of knowledge. This is not only useful for this work, but will also benefit my future professional careers.

Dr. Renaut and Dr. Gelb I thank for their organizing research activities in image sciences at ASU. The medical imaging conference and research seminars they organized were really helpful to learn the imaging applications. I take advantage of their connection with researchers in hospital to collect clinic MR data which were used in my dissertation.

IPAM at UCLA and MSRI at UC Berkeley had organized workshops in numerical optimization and its applications. The talks and discussions with other researchers enriched my research tremendously.

Finally I thank my family, friends and the people who offered me great support during my studies and the work on this dissertation.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

In this work, we generally focus on the inverse problem using sparse regularization, especially in the application to sparse MR imaging and the source localization problems. We present a new approach based on the sparse representation paradigm. The purpose of this chapter is to introduce the problems addressed in the thesis, motivate the need for a new approach, and describe our main contributions as well as the organization of the paper.

1.1 Overview of the problems addressed in the thesis

The core of this work is a numerical scheme for solving a large scale sparsity enforcing regularized inverse problem and its applications in both sparse MR imaging and source localization problems.

Inverse problems have a wide range of important practical applications in the areas of signal/image processing including radar imaging, digital photography, astronomic imaging, topographic imaging, etc.[A.K91][HB77][MP98]. Image restoration is one of the earliest and most classical linear inverse problem which dates back to the 1960s [HB77]. The goal is to recover an image from a small number of linear measurements. In many fields of science and technology, one can only collect a limited number of measurements about an object of interest, because of some physical constraints on the equipment or the highly cost in collecting the full data sets. That means we have to use a small proportion of data to estimate the overall data, which indicates a linear inverse problem. For instance, recovering a single MR image commonly involves collecting a series of data, called acquisitions, and reconstructing the image by solving an inverse problem with certain regularization. In the acquisition, a strong magnetic field and a radio frequency (RF) pulse are directed to a section of the anatomy,

causing the protons to become aligned along the magnetic direction and spin with a certain frequency. After the RF is turned off and the protons return to their natural state, a RF signal is released and captured by the external coils in the form of phases. That is, the data is collected along a particular trajectory, such as straight lines from a Cartesian grid, in spatial frequency space or *k*-space. We can just reconstruct the image from such acquisitions by the inverse fast Fourier transform (IFFT). Traditionally, the *k*-space sampling pattern is designed to meet the Nyquist criterion: the number of samples needed to reconstruct the image without error is dictated by its bandwidth. Image resolution depends on the size of the sampled region of *k*-space and the supported field of view depends on the sampling density within the sampled region, so when the number of the sampling violates the Nyquist rate, artifacts appear in the process of the linear reconstruction.

In many practical MR image applications, the sampling speed is fundamentally limited by the physical constraints of the equipment and usually the scanning is a long and uncomfortable process. Many researchers are striving to reduce the amount of acquired data without degrading the image quality. The recently proposed theory of *Compressive Sensing* (CS) [D.L06][E.C06] is a good approach to reduce the redundancy of the required MR data. A successful application of CS is composed of two key steps: the encoding process and the decoding process. In the encoding process, the underlying image must have a sparse representation under a known transform domain; in the decoding process, the image is recovered by solving a nonlinear optimization model which can preserve the sparsity of the image and the consistency of the reconstruction with the sampled data. Aliasing artifacts caused by *k*-space undersampling must be incoherent in the sparse transform domain. Some natural properties of MRI make it fit the assumptions of CS theory very well. As we know, natural images can often be compressed with little loss of information [DM02], and medical images are also compressible, such as the JPEG-2000 standard. Some MR images are simply sparse

in pixel domain, such as angiograms, which are the images of contrast enhanced blood vessels in the body. Furthermore, the sparsity of general MR images is evident by the fact that the images can be represented under an appropriate transform domain by just a few large coefficients and many small coefficients [S.M99], which indicates a well approximated sparsity. In this context, we will focus on some popular transform encoding operators in MR imaging, such as wavelet, finite difference, etc. Sparsity is a valuable property to have and we pose the MR image reconstruction as a linear inverse problem with sparsity enforcing regularization. The observed data undersampled in *k*-space in terms of an overcomplete basis is not unique and we impose a penalty to regain the uniqueness, and more importantly, to obtain the solution with sparse structure.

The ideal penalty to enforce sparsity is to minimize the number of nonzeros in the underlying spectrum (which is referred to as the ℓ_0-norm of the spectrum). However,the resulting problem is combinatorial in nature, and generally NP-hard [B.K95]. We use a more tractable ℓ_1-norm penalty instead, which is well known in signal processing and has been proposed as a convex alternative to the ℓ_0-norm. Actually, the solution of a noiseless signal representation problem using ℓ_0 penalty has a close connection to solutions using ℓ_1 penalties. In the ℓ_1-norm, many small coefficients tend to carry a larger penalty than a few large coefficients, therefore small coefficients are suppressed and the sparsity is preserved. Solving ℓ_1-norm regularized inverse problems is much simpler, but this does not mean trivial, since the image processing problem is of a large scale in general and usually a strong background noise is present in the undersampled frequency data. We use a quadratic term to preserve the consistency and do the denoising.

In the MR imaging application, other types of sparse transformations and their numerical solutions are of our interest. For instance, the *Total Variation* [LE92] operator has been widely used in image deblurring, and its numerical solution has a long term interest because of its high nonlinearity and non-differentiability in the compu-

tation. We want to adapt the MR image reconstruction problem to a CS application by following the assumptions of CS. And the algorithm for minimizing the inverse problem with ℓ_1-norm or other nonlinear regularization should be both accurate and efficient. However, to balance this tradeoff for the large scale problems is still a challenge.

In this thesis, we propose an algorithm based on the augmented Lagrangian method which is fast and robust to the regularization parameters. In our framework, we first introduce the new variables into the original sparsity enforcing operator-regularized inverse problem to get an approximated problem which can be solved by using the two dimensional shrinkage formula, and then form a quadratic problem to penalize the discrepancy between the original terms and the approximated terms, as well as the fidelity term. In each iteration, the solution of the original problem is calculated from two subproblems utill the final result satisfies the stopping criteria. The main computation is in the process for solving a conjugate gradient routine and we can apply a preconditioner to accelerate it by taking advantage of the block Toeplitz structure of the iteration matrix.

In the second part of this work, we mainly focus on adapting our proposed sparsity enforcing algorithm to solve the inverse problem with multiple measurements, especially derived from the problems of detecting the source localization using sensor arrays [IB97][MA05]. As a natural extension of the single measurement case, the sparse reconstruction based on multiple measurement data has wide application in signal processing. When the data are not sufficient to make a convincing estimation, the easy and cheap way to increase the volume of the data is to keep collecting from the temporal domain, instead of from the spatial domain due to the physical limit of the underlying object or some constraints of the equipment. In this case, the measurement data is a time series and finally our estimation can be made based on more information than only from the spatial domain. Generally, we use this procedure to increase the

volume of data in the problem of detecting the source location using sensor arrays.

The source localization methods have been actively investigated for years, and they play a fundamental role in many applications, such as acoustic, electromagnetic and seismic sensing. The goal of source localization methods is to be able to find the location of closely distributed sources from noisy data collected from the sensor array. To improve the estimation performance and the robustness of the sensor network, in the presence of noise over classical maximum likelihood and subspace methods, sparsity based localization have been slowly gaining popularity [MA05][DM06][IB97] [VG08][VR08]. The localization problem can be formulated as the sparse approximation of the measured signals in a specific dictionary of atoms, which is produced by discretizing the space with a localization grid and then synthesizing the signals received at each sensor from a source located at each grid point. Since the possible number of sources is much less than the size of the relative discretized localization grids, it is reasonable to view the structure of the possible localization of the sources as sparse in terms of the localization grids. In this context, the search of the sparsest approximation to the received signals that minimizes the data error implies that the received signals were generated by a small number of sources located within the localization grid. Hence our algorithm detects the location of the sources successfully by exploiting the relationship between the small number of sources present and the corresponding sparse representation of the received signals.

As in numerous non-parametric source localization techniques, we estimate the energy of the signal as a function of location and this perfectly contains the dominant peaks of the sources at the place where they are detected by the sensor array. We exploit the signal field through the sensor observation which is obtained through the sensing matrix synthesizing the known information: the geometry of the sensor array, the parameters of the medium where the signal propagates, and the measurements of the sensors. So the sparse signal structure in the underlying spatial spectrum can be

reconstructed from solving an ill-posed inverse problem by applying proper sparsity enforcing regularization. In this thesis, we first show how to formulate the source localization problem into an inverse problem using a sparsity enforcing regularization framework. Then we propose an efficient and robust sparse reconstruction algorithm for detecting the source locations from single time snapshot processing to multiple time snapshot processing. A series of theoretical analysis is constructed in order to guarantee the convergence of the algorithm. We conduct several numerical experiments to make a comprehensive comparison between the proposed algorithm and some other current frameworks for source localization.

1.2 Outline and contributions

Before we introduce the content of this thesis chapter by chapter, we want to briefly summarize our main contributions. The first contribution in this thesis is the development of a sparse reconstruction framework for ℓ_1-norm and *Total Variation* regularized inverse problems. In this framework, we reformulate the objective function via the augmented Lagrangian methods such that the original large scale problem can be separated into several subproblems which can be solved alternatively via 2D shrinkage and conjugate gradient methods. We adapt the model and apply it to the MR imaging; we argue that the model matches the assumptions of compressive sensing theory very well. We discuss the incoherence between the different sampling patterns and the sparse transformation, as well as the relationship between the sampling density and the reconstruction error. An efficient preconditioning scheme and the convergence analysis of the algorithm as well as the optimal parameter selection are discussed in depth; sufficient numbers of numerical simulations are conducted, in order to widen the range of the applications of the proposed algorithm in MR imaging and for testing its robustness. The processing time and reconstruction accuracy is greatly enhanced. In the second part, we focus on another application of the sparse reconstruction in the problem of source localization with sensor arrays. We reformulate our algorithm to fit

the data sampled from both the spatial domain and the temporal domain, but we only pursuit the sparsity in the spatial domain. The results of several numerical experiments are included for comparing our routine with some other current packages.

Chapter 2: A Brief Survey of Existing Sparse Optimization Algorithms We start by giving an overview on the general discrete ill-posed inverse problem and motivate the need for the regularization as well as show its shortcomings in enhancing the sparsity. We introduce the recently developed theory of Compressive Sensing which motivates a large amount of research in sparse optimization algorithm development. Then a summary on several current popular sparse optimization algorithms is given.

Chapter 3: The TV-ℓ_1 Sparsity Enforcing Algorithm In this chapter, we first reformulate the TV-ℓ_1 objective with linear equality constrain into a separable structure. Then a fast multi-splitting algorithm based on the augmented Lagrangian method is presented. An in depth discussion of theoretical issues including the optimality of the algorithm, the convergence rate, the effects of the parameters on the convergence and the stopping criteria are given in the following. Furthermore, a preconditioner for a BTTB matrix is presented for accelerating the proposed algorithm.

Chapter 4: Numerical Results and Application to MRI We mainly present the numerical experiments and show the performance of the proposed algorithm. First, we test the robustness of the proposed algorithm w.r.t. regularization parameters, show how the parameters affects the convergence rate and some practical issues on implementing the BCCB preconditioner. Next, we show the basic MR imaging principles and the application of CS in sparse MRI, then the numerical comparison with other existing packages for reconstructing the real MR images are presented in the following.

Chapter 5: Source Localization Detection with Sparse Reconstruction This chapter is devoted to the analysis of the techniques in developing methods for source location detection. We first reformulate the sensor output into a sparse representa-

tion under a modified overcomplete basis. The advantages of the TV-ℓ_1 model and the ℓ_p norm in detecting the source location is discussed. Then an efficient algorithm for minimizing a non-convex objective is presented. In the last part, the numerical results show that the proposed algorithm has better performance in resolving closely distributed sources than existing methods.

Chapter 6: The future work We mainly summarize this work and point out some of our future approaches for extending the current work.

Chapter 2

A Brief Survey of Existing Sparse Optimization Algorithms

In this chapter, we consider the general linear ill-posed inverse problems of solving the underdetermined system $Ax = b$ and the optimization algorithms for reconstructing sparse objectives. The uniqueness of the solution of an underdetermined system relies on a regularization term that is determined by prior information on the underlying objectives. On the one hand, although the traditional Tikhonov regularization is well known and has been widely used in image processing, it shows that it is not suited for preserving the sparse feature of the objectives; on the other hand, since the theory of Compressive Sensing was proposed, the optimization algorithms for solving the ℓ_1-norm related problems receive much attention. In this context, our discussion covers the regularized inverse problems and the current progress on the optimization methods for reconstructing the sparse or transformed sparse objectives.

2.1 The ill-posed inverse problem and regularization

Since the theory of Compressive Sensing (CS) was proposed in 2005, it attracts much attention in signal processing and optimization communities. The CS theory states that a minimum ℓ_1-norm solution to an underdetermined linear system is the sparsest possible solution under quite general conditions. Specifically, suppose $x \in \mathbb{R}^N$ is an unknown signal, $b \in \mathbb{R}^M$ is the measurement vector ($M < N$), and the measurement matrix $A \in \mathbb{R}^{M\times N}$ is of full rank. Generally, the underdetermined system $Ax = b$ forms a linear inverse problem. If x is sufficiently sparse and the sensing matrix A is incoherent with the basis under which the signal x has a sparse representation, then x can be reconstructed from a much smaller measurement b via minimizing the ℓ_1-norm of x such that it satisfies the underdetermined system $Ax = b$.

Traditional ways of solving the linear inverse problem $Ax = b$ are by linear

least squares, in which, one finds the minimum ℓ_2-norm solution to the system, or by Tikhonov regularization. Since the solution of the underdetermined system is not unique, to solve for x from a finite number of measurement y can be approached via the singular value decomposition (SVD) of the measurement matrix A. Suppose $A \in \mathbb{R}^{M\times N}$ is full rank. Its SVD can be represented as:

$$A = U diag(\sigma_i) V^T = \sum_{i=1}^{m} u_i \sigma_i v_i^T, \tag{2.1}$$

where u_i and v_i^T are the i-th column of the orthogonal matrix U and V^T respectively, σ_i are the singular values of A aligned in the diagonal matrix $diag(\sigma_i)$ and ordered as a dereasing of its magnitude. Then the solution x can be represented via the Moore-Penrose pseudo inverse as:

$$x_{true} = A^\dagger b = \sum_i v_i \sigma_i^{-1} u_i^T b = \sum_i \frac{u_i^T b}{\sigma_i} v_i. \tag{2.2}$$

However, instability arises from dividing by the small singular values. In practice, the observation measurements often involve strong background noise. Mathematically the underdetermined system with additional noise can be rephrased as

$$b = Ax + n, \tag{2.3}$$

where n denotes the additional noise. In this case, theoretically the solution to this system can be represented via pseudo inverse as

$$\begin{aligned} A^\dagger b &= x_{true} + A^\dagger n \\ &= x_{true} + \sum_i \frac{u_i^T n}{\sigma_i} v_i, \end{aligned} \tag{2.4}$$

because of the randomness in the noise and the division by small singular values, the last term in the second line of the above formula becomes unbounded, and the solution becomes highly sensitive to perturbation in the error term. To overcome the ill-posedness in inverse problem, filters are widely used to counterbalance the effects

of the small singular values to the solution. For instance, the Tikhonov filter function [A.N63b][A.N63a] is given as:

$$\omega_\lambda(\sigma_i) = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}. \tag{2.5}$$

Then plugging (2.5) into (2.2), the filtered solution can be expressed as:

$$\begin{aligned} x_\lambda &= \sum_i v_i \omega_\lambda(\sigma_i)\sigma_i^{-1} u_i^T b = \sum_i \frac{\sigma_i (u_i^T b)}{\sigma_i^2 + \lambda} v_i \\ &= (A^T A + \lambda I)^{-1} A^T b, \end{aligned} \tag{2.6}$$

where the positive parameter λ is called the regularization parameter. It determines the threshold level for the Tikhonov filter. The Tikhonov filtered solution x_λ given in (2.6) is equivalent to the minimizer of ℓ_2-norm regularized least squares problem:

$$\min_x \frac{1}{2}||Ax - b||^2 + \lambda ||x||^2 \tag{2.7}$$

The selection of the parameter λ controls the tradeoff between the noise level and feasibility of the solution in the ell_2 ball. So if its value is too small, the solution x_λ becomes highly sensitive to the noise. The filtering is not adequate. On the other hand, if λ is a large value, the noise term will be filtered out and some components of the solution will also be cut off at the same time. Selection of the regularization parameter is essential. Many methods have been proposed, such as, the L-Curve method, discrepancy principle, generalized cross validation (GCV) and many other methods based on the statistics of the background noise. However, the selection of the regularization parameter is still an open problem, especially when the objective function is nonlinear or the statistics of the background noise is unknown. In the following chapters, we will give a detailed discussion on this issue from both the aspects of the effects of the parameter value to numerical performances and the robustness of the solutions.

For large scale problem, it is often not practical to solve for the solution via SVD since it requires a large matrix, and is numerically inefficient. The alternative

variational representation of the Tikhonov regularization (2.7), equivalent to (2.6), is much easier to solve. The regularization parameter λ in (2.7) balances the noise level and the fitness of the data, and here we want to find a solution to the undetermined system with minimum ℓ_2-norm via minimizing the ℓ_2-norm regularized problem (2.7). This method has been widely used in image denoising and proved to be efficient. But it is worth to point out that using the ℓ_2-norm tends to penalize the large entries in x more than the smaller ones, so the ℓ_2-norm is not suited for the underlying objective with sparse structure, such as the spike data in geophysics [SP81]. Alternatively, the ℓ_1-norm regularization is getting more attention because of its good properties in enhancing the sparsity and numerical tractability. In our work, we mainly work with two regularization terms: the ℓ_1-norm and total variation (TV). In the following, we will discuss the total variation regularization, and place the discussion on ℓ_1-norm regularization in the next section together with the introduction of idea of sparse optimization and Compressive Sensing.

The total variation (TV) is first introduced into image processing by Rudin, Osher and Fatami (ROF) [LE92] in 1992. A discrete version of unconstrained ROF model can be expressed as a TV regularization term plus a ℓ_2-norm fidelity term:

$$\min_x \frac{\lambda}{2}||Ax-b||^2 + TV(x). \tag{2.8}$$

Although the ROF model is first introduced for image denoising, this methodology can be easily extended to restore blurred images by adapting A in (2.8) into a known linear blurring kernel. Over the years, it has been widely used and proved to be successful in dealing with image denoising and deblurring problems, image imprinting problem and image decomposition problems as well as CT and MR imaging. The main advantage of the TV formulation is its ability to preserve sharp edges of the image, due to its piecewise smoothness property. Generally, the TV norm is defined as the sum of the Euclidean norm of the finite differences of each pixel in the underlying image.

We assume that the image domain Ω is square, and define a regular $N \times N$ grid of pixels, indexed as (i, j), for $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, N$. The images can be represented as two-dimensional matrices of dimension $N \times N$, where $u_{i,j}$ represents the value of the function u at pixel (i, j). To define the discrete total variation, we introduce a discrete gradient operator, whose two components at each pixel (i, j) are defined as follows:

$$D^{(1)}u = \begin{cases} u_{i+1,j} - u_{i,j} & i < N, \\ 0 & i = N. \end{cases} \tag{2.9}$$

$$D^{(2)}u = \begin{cases} u_{i,j+1} - u_{i,j} & i < N, \\ 0 & i = N. \end{cases} \tag{2.10}$$

So $D^{(1)}$ and $D^{(2)}$ are $N^2 \times N^2$ matrices. If the 2D difference operator D is denoted as $D = [D^{(1)}; D^{(2)}]$ and the square image u is vectorized as a column vector, then the discrete TV of the image u is defined as:

$$TV(u) = \sum_i ||Du||, \tag{2.11}$$

where $||\cdot||$ is the Euclidean norm, and the index i goes through all pixels of u. We use this notation of TV in all through out this work.

One of the major reasons for the ongoing research into TV deblurring problems is that the non-differentiability and the non-linearity of the TV norm makes it difficult to find a fast numerical method. The first order derivative of the TV norm involves the term $\frac{\bigtriangledown u}{|\bigtriangledown u|}$, and it is degenerate when $|\bigtriangledown u| = 0$. Currently, a number of numerical methods have been proposed for unconstrained TV denoising or deblurring models, and they include partial differential equation based methods, such as explicit [LE92], semi-implicit [DXC06] or operator splitting schemes [MXC04], and fixed point iterations [CM96]. Optimization oriented techniques include Newton-like meth-

Figure 2.1: From left: (1) The 256 by 256 Cameraman; (2) The Isotropic total variation norm of the 'Cameraman'.

ods [TK06][KK99][YF96][TP99], second order cone programming [DW05], interior-point methods [EJ05][HJ06], and conjugate gradient methods [MJ07a]. In this work, we propose a fast algorithm based on the method of augmented Lagrangian multipliers for minimizing the TV and ℓ_1-norm regularized inverse problem as well as its application in sparse MR imaging. In the proposed algorithm, we split the TV norm into several subproblems with closed form solutions and process each one in parallel such that in this way the proposed algorithm is much faster than most of the existing methods.

2.2 The sparse optimization and compressive sensing

Our main objective is to find a sparse solution to an underdetermined inverse problem, that is the solution with minimum number of nonzero components. Mathematically the problem can be expressed as:

$$\begin{aligned} \min_{x} \quad & ||x||_0 \\ s.t. \quad & Ax = b, \end{aligned} \tag{2.12}$$

where $A \in \mathbb{R}^{M\times N}$ is a full rank matrix with $M \ll N$, the l_0-norm $||\cdot||_0$ counts the number of nonzero entries and $b \in \mathbb{R}^M$ is the given observation which may or may not involve

additional background noise. Apparently this problem can be solved in finite time. Denote $A = [A_1, \cdots, A_N]$ with each A_i representing the i-th column vector in A. We can form a sequence of square matrices $A_T \in \mathbb{R}^{M \times M}$ by exhausting any combinations of M linearly independent columns of A. Then solve for z from each linear system $A_T z = y$ and set the one with the smallest number of nonzero entries as the sparsest solution to the linear system (2.12). Theoretically for finding the sparse solution, we could have to solve N choose M ($\binom{N}{M}$) linear equations at most. However this way is computationally impractical, since the quantity of $\binom{N}{M}$ grows exponentially fast as $N, M \to \infty$. For instance, for the problem of $N = 2M = 1024$, it is necessary to solve 2^{512} linear systems of 512×512, which can not be done using current computing tools. So (2.12) can not be solved within polynomial time and is a NP-hard problem.

Alternatively, we may consider to solve the problem (2.12) in another way using the ℓ_1-norm. Minimizing ℓ_1-norm now attracts more and more attention for its tractability in computation, and this problem can be represented as:

$$\begin{aligned} \min_{x} \quad & ||x||_1 \\ s.t. \quad & Ax = b, \end{aligned} \tag{2.13}$$

where $||x||_1 = \sum_{i=1}^{N} |x_i|$ for $x = (x_1, \cdots, x_N)^T$, and the ℓ_1-norm minimization problem is equivalent to a linear programming and compared to the ℓ_0 problem, (2.13) is more computationally tractable. The above model is also called basis pursuit (BP) problem [S.S99]. Unlike the energy norm minimization problem which tends to penalize the large components more, the minimizing the ℓ_1-norm preserves the large components while penalize the smaller entries much more so that the sparsity structure of the underlying objective can be enhanced. Applying the ℓ_1-norm to restore the sparse objective has been well known since 1970's when it was first applied to restore the spike train signal in geophysics. Since then the ℓ_1-norm is widely used in signal processing. Recently, the theory of Compressive Sensing extends its applications and some theo-

retical issues are also addressed. As a potential alternative of ℓ_0-norm are considered, one of the most important question to be clarified is under what conditions the solution to (2.13) is unique and equivalent to the solution to (2.12).

There are mainly two types of concepts for depicting the properties of sensing matrix A in order to state the situation of the equivalence of the l_0-norm and ℓ_1-norm, one is the *mutual coherence* (MC), the other is the *restricted isometric properties* (RIP). Donoho and others provide ground-breaking work and a show a series of papers [D.L06][DA92][D.L95][DM03][DX01] are the exact conditions of the equivalence of the l_0 and ℓ_1 minimization using the concept of MC. It states that for the case with sensing matrix $A \in \mathbb{R}^{N/2\times N}$ obtained by concatenation of two orthonormal bases, the solutions to both (2.12) and (2.13) are unique and identical provided that in the most favorable cases, the sparsity level K (# of nonzero entries) of the vector x is at most $.914N/2$. Candes, Tao and Romberg [E.C05][E.C06][E.C04] use a very different way and proved the equivalence holds with overwhelming probability for various types of random matrices provided that the number of nonzero entries K in the underlying vector x be of the order of $N/logN$ with the aid of the concept of RIP. For the sake of completeness of our discussion, we simply recall some of the key points in the theory developed by Candes.

Let $0 < K < M$ be an integer and let the submatrix A_T be obtained by extracting the columns of A corresponding to the indices in $T \subset \{1,2,\cdots N\}$. Then the K restricted isometry constant δ_K of A is the smallest quantity such that

$$(1-\delta_K)||x||_2^2 \leqslant ||A_T x||_2^2 \leq (1+\delta_K)||x||_2^2 \tag{2.14}$$

for all subsets T with Cardinality $card(T) \leqslant K$. Hence if a matrix A has such a constant $\delta_K > 0$ for some K, then A possesses the RIP. This property essentially requires that every set of columns with cardinality less than K approximately behaves like an orthonormal system, and if the sensing matrix A is RIP, then exact recovery is possible.

When sensing matrix A possesses the RIP, it is very easy to show that under the condition $\delta_{2K} < 1$ then the K sparse solution of (2.13) is unique. Actually if the solution to (2.13) is not unique, we can assume both x^1 and x^2 are its solution, that is the discrepancy satisfies that

$$A(x^1 - x^2) = 0, \tag{2.15}$$

apparently $(x^1 - x^2)$ has $2K$ nonzeros at most. We can choose the index set T containing the indices of the nonzero entries in $x^1 - x^2$ such that

$$(1 - \delta_{2K})||x^1 - x^2||_2^2 \leq ||A_T(x^1 - x^2)||_2^2 = 0, \tag{2.16}$$

for all subsets of T, then (2.16) implies the uniqueness of the solution. The following theorem given by Candes, Romberg and Tao in 2006 gives the conditions on the sensing matrix implying the uniqueness of the solution and equivalence of the l_0 and ℓ_1 minimization.

Theorem 2.2.1. (Candes, Romberg and Tao 06)
Suppose that $K \geqslant 1$ is such that

$$\delta_{3K} + 3\delta_{4K} < 2 \tag{2.17}$$

and let $x \in \mathbb{R}^N$ be a vector with $||x||_0 \leq K$. Then for the inverse problem $Ax = b$, the solution of (2.13) is unique and equal to x.

The prove of the above theorem is given in [E.C06] and we omit it here. But it is worth to point out that although the above theorem explains when the solution to problem (2.13) and (2.12) are equivalent and has already been used as an fundamental theorem for reconstructing the sparse signals using ℓ_1-norm minimization, there is no explicit construction of matrices of any size that possess the RIP. Candes, Romberg and Tao proved that a matrix with RIP can be found with positive probability as long as $M > cKln(N(1+2/\varepsilon)/e)/\varepsilon^2$.

Theorem 2.2.2. (Candes, Romberg and Tao 06)

Suppose the random matrix $A = [a_{ij}]_{1\leq i\leq M, 1\leq j\leq N}$ *is iid with mean zero and variance* $1/\sqrt{m}$*, then the probability that A possesses RIP:*

$$\textbf{Prob}(|||Ax||_2^2 - ||x||_2^2| \leqslant \varepsilon||x||_2^2) \geqslant 1 - \binom{N}{K}(1+2/\varepsilon)^K e^{-M\varepsilon^2/c} \tag{2.18}$$

for any vector $x \in \mathbb{R}^N$ *with* $||x||_0 = K$*, where* $c > 2$ *is a constant.*

We need to point out that since $\binom{N}{K} \leqslant (N/e)^K$, then

$$\binom{N}{K}(1+2/\varepsilon)^K e^{-M\varepsilon^2/c} \leqslant e^{-M\varepsilon^2/c + Kln(N/e) + Kln(1+2/\varepsilon)},$$

so when $M > cKln(N(1+2/\varepsilon)/e)/\varepsilon^2$, we have

$$Prob(|||Ax||_2^2 - ||x||_2^2| \leqslant \varepsilon||x||_2^2) > 0,$$

and the probability of a matrix possessing RIP is positive. The RIP can also be verified using the MC of the sensing matrix A defined as

$$\chi_A = \max_{1\leq i,j\leq N, i\neq j} |[A^T A]_{ij}|, \tag{2.19}$$

D.Donoho [DM03] points out that the RIP constant $\delta_K \leqslant \chi_A(K-1)$ via the Gershgorin circle theorems, and it is common that when $MC(A) \simeq \frac{1}{\sqrt{M}}$, we have the non-trivial RIP bounds for $K \simeq \sqrt{M}$. Unfortunately, no known deterministic matrix yields a substantially better RIP. The RIP holds for Gaussian and Bernoulli matrices, when $K \simeq M/log(N/M)$; for more structured matrices, such as the random section of discrete Fourier transform (DCT), RIP often holds when $K \simeq M/(logN)^p$ for a small integer p [E.C04]. This fact explains the benefit of randomness in ℓ_1 compressive sensing.

The ℓ_1-norm is well known in preserving the sparse features, and some of its early application can be found in the area of geophysics [HJ79][JF73][SW86][SP81] where sparse spike train signals with large sparse errors are of interest. In the last two decades much research has been aimed at finding a sparse solution of an ℓ_1-norm

regularized inverse problem, and its applications are extended to many areas, such as the wavelet based image deconvolution and reconstruction, the least absolute shrinkage and selection operator (LASSO), the low rank matrix approximation and compressive sensing. We will show some examples motivating the research of sparse optimization.

Example 1: Sparse Signal Reconstruction

The most direct application of ℓ_1 minimization is the reconstruction of a sparse signal. Suppose $y \in \mathbb{R}^M$ represented via a tight frame (orthonormal matrices) $A \in \mathbb{R}^{M\times N}$ ($M << N$) is an observation of unknown sparse signal $x \in \mathbb{R}^N$. One reconstructs x via solving the problem:

$$\min\{||x||_1, s.t. ||Ax-y|| \leq \varepsilon\}, \tag{2.20}$$

where $\varepsilon > 0$ represents the noise level of the observation.

Example 2: Low Rank Matrix Approximation

The low rank matrix approximation problem arises from the principle component analysis (PCA) having wide range of applications in the engineering and statistics, where one tries to use a matrix with lower rank to approximate an original data matrix without affecting the fitness of the data. Mathematically, suppose $D \in \mathbb{R}^{M\times N}$ represents the collection of the data and the data in each column of D represents a property of the objective, then we try to find a low rank matrix A such that the discrepancy is minimized, which leads to the problem:

$$\begin{aligned} \min_{A,E} \quad & ||E||_F \\ s.t. \quad & rank(A) \leq r, \\ & D = A + E. \end{aligned} \tag{2.21}$$

where $||\cdot||_F$ is the Forbenius norm corresponding to the assumption that the data are corrupted by the Gaussian i.i.d. noise. $r \leq \min\{M,N\}$ is the target dimension of the subspace. However, one still needs a way to efficiently and accurately recover A from

a corrupted data matrix $D = A + E$, since in model (2.21), some entries of the additive errors E may be arbitrarily large and may lead to A being far from the true value.

Recently, [EB08][JY09] show that the exact recovery of A is achievable as long as the noise matrix E is sufficiently sparse with respect to A, by solving the following convex optimization problem:

$$\begin{aligned} &\min_{A,E} ||A||_* + \lambda ||E||_1 \\ &s.t. \quad D = A + E, \end{aligned} \tag{2.22}$$

where the nuclear norm $||\cdot||_*$ represents the sum of the magnitudes of the singular values, and λ is a positive weighting parameter for balancing the ℓ_1-norm and nuclear norm. Due to the ability to exactly recover underlying low-rank structure in the data, even in the presence of large errors or outliers, this optimization is referred to as robust PCA (RPCA), and it has been widely used in background modeling and for removing shadows, peculiarities from face images, etc.

Example 3: Wavelet based Image Reconstruction

Most natural or man-made images are compressible through a well defined basis, that is the coefficients of the signal represented by an appropriate basis possess only a few large components and others are close to zero. In this context, if we threshold the small coefficients, the overall quality of the image will not be damaged. The way of representing the image in a new basis to have sparse coefficients is the so called transform encoding [S.M99]. For example, the JPEG2000 standard 2 uses the fact that the representation of natural images using Daubechies [I.D92] maxflat wavelet bases is considerably sparser than the original representation.

This application benefits from the sparse transform using wavelet. Suppose an image $u \in \mathbb{R}^{N\times N}$ has a sparse representation under the transformation Ψ which can be one involving wavelet transforms or a redundant dictionary. In this context, the coefficients of the unknown image $x = \Psi^T u$ are sparse under this basis. Then the

sensing matrix has the form $A = \Theta\Psi$, where Θ is an observation operator which could be a blur kernel, a tomorgraphic projection or Gaussian random projection or others. Hence, the original image u can be reconstructed by solving the problem:

$$\begin{aligned} \min_{u} \quad & ||\Psi^T u||_1 \\ s.t. \quad & ||\Theta u - b||^2 \leq \delta, \end{aligned} \tag{2.23}$$

where $\delta > 0$ is the noise level, and in this application TV regularization can also be involved.

□

Compressive Sensing (CS) is a popular new application utilizing sparse optimization using ℓ_1-norm. Recent results show that a relatively small number of random projections of a sparse signal can contain most of its salient information. Accurate approximations can be obtained by finding a sparse signal that matches the random projections of the original signal. Generally speaking, there are two main steps in CS, the so called encoding and decoding. In the encoding step, one allocates a $M \times N$ ($M << N$) linear transformation Φ to the underlying unknown $x \in \mathbb{R}^N$ such that the information about the unknown x is compressed in a data vector $y = \Phi x$ whose dimension is relatively much smaller than that of the unknown x. In the decoding step, let $\triangle$ denote a decoder which is usually a nonlinear transformation, then an approximation $\tilde{x}$ provided by the decoder $\triangle$ can be expressed as $\triangle y = \tilde{x} \approx x$. These encoding and decoding steps lead to an economical way of recording the information and restoring the unknown by using prior knowledge and partial data. The core question in CS is to find an appropriate pair of encoder and decoder $(\Phi, \triangle)$ such that the approximation $\tilde{x}$ fits the accuracy. A series of research paper [D.L06][DA92][E.C05][E.C06][E.C04] have shown that when the sensing matrix meets a quite general condition, the decoder that is related with minimizing a ℓ_1-norm related problem (2.13) is also the sparsest possible solution to the underdetermined system.

The ℓ_1 minimization problem (2.13) can be recast into a linear program (LP) and conventional methods, such as interior point methods, are applicable. However, the computational complexity of these general-purpose algorithms is too high for many real world large-scale applications. Alternatively, motivated by finding a more efficient algorithm for solving the problem, many new algorithms have been proposed, such as Gradient Projection (GP), Homotopy, Iterative Shrinkage-Thresholding (IST), *ect*. The main contribution of our thesis is that we propose a fast algorithm based on the Augmented Lagrangian Multiplier (ALM) for solving the ℓ_1 and TV regularized problem as well as its real world application in sparse MRI. In the following, we summarize some existing algorithms and compare them from different prospectives.

2.3 Review of some existing sparse reconstruction algorithms

The traditional algorithms for solving the ℓ_1-norm sparse optimization problem tend to be slow in large scale CS application. This is mainly because the sensing matrix that is composed of random matrices and matrices whose rows are taken from orthonormal matrices, such as a partial Fourier matrix, are invariably dense. Besides, because of the size and density of the data involved, one should take advantage of the techniques needing only a matrix vector multiply, instead of a matrix factorization. In practice, many natural or man made signals are compressible with respect to dictionaries constructed using principles of harmonic analysis, such as the wavelet. So this type of structured dictionary often comes with a fast transformation algorithm. Thus it is necessary to develop an algorithm which is fast and robust for the compressed sensing signal reconstruction. In this section, we provide an overview of some existing algorithms for solving the ℓ_1-norm minimization problem (2.13).

2.3.1 Primal dual interior point methods

The Primal Dual Interior Point Methods (PDIPM) [N.M89][N.K84][RI84] is a standard way for solving linear programs. As the ℓ_1-norm minimization problem (2.13) can be

recast as a linear programming, the PDIPM becomes a natural choice for solving this problem. Suppose under usual standard assumptions, (2.13) is converted into a standard linear program denoted as the primal problem (P) below:

$$\begin{aligned} (\mathbf{P}) \quad & \min_{x} \quad c^T x \\ & s.t. \quad Ax = b, \quad x \geqslant 0, \end{aligned} \tag{2.24}$$

where for ℓ_1 minimization $\mathbf{c} = \mathbf{I} \in \mathbb{R}^N$, and its dual problem is given as:

$$\begin{aligned} (\mathbf{D}) \quad & \max_{y,\, z} \quad b^T y \\ & s.t. \quad A^T y + z = c, \quad z \geqslant 0, \end{aligned} \tag{2.25}$$

where $y \in \mathbb{R}^M, z \in \mathbb{R}^N$ are the dual variables. The PDIPM updates the variable (x, y, z) via solving the $(\mathbf{P})$ and $(\mathbf{D})$ simultaneously.

Let us assume the problem is strictly feasible, this means that there exist dual variables $y \in \mathbb{R}^M, z \in \mathbb{R}^N$ and $x \in \mathbb{R}^N$ satisfying the **KKT** condition as below:

$$F(x,y,z) = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ \mathbf{XZe} \end{bmatrix} = 0, \tag{2.26}$$

where $\mathbf{X} = diag(x_1, \cdots, x_N)$, $\mathbf{Z} = diag(z_1, \cdots, z_N)$ and $(x, z) \geqslant 0$. Then Newton's method forms a linear model for $F(x,y,z)$ and the search direction $(\triangle x, \triangle y, \triangle z)$ can be generated as below:

$$J(x,y,z) \begin{bmatrix} \triangle x \\ \triangle y \\ \triangle z \end{bmatrix} = -F(x,y,k), \tag{2.27}$$

where $J(x,y,z)$ is the Jacobian of F. For the strictly feasible current point (x,y,z), the Newton equation (2.27) can be expanded as:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \triangle x \\ \triangle y \\ \triangle z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XZe \end{bmatrix}. \tag{2.28}$$

One performs a line search along the Newton direction so that the new iterate is

$$(x,y,z) + \alpha(\triangle x, \triangle y, \triangle z). \tag{2.29}$$

In practice, it is very hard to find a strictly feasible starting point. So we may consider to relax the feasibility and linear complementarity condition and improve them step by step, and this leads to the infeasible interior point methods, which only require the components of the initial points (x^0, z^0) to be strictly positive. To improve the feasibility in each iteration, we can use the complementary slackness condition $x_i z_i = 0$ to $x_i z_i = \tau$, and set the primal, dual, and central residuals quantifying how close a point (x,y,z) is to satisfy the **KKT** (2.26):

$$r_{pri} = Ax - b,$$

$$r_{dual} = A^T y + z - c,$$

$$r_{cent} = XZe - \tau e,$$

An inexact Newton direction can be generated from the equation given by:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \triangle x(\tau) \\ \triangle y(\tau) \\ \triangle z(\tau) \end{bmatrix} = \begin{bmatrix} -r_{dual} \\ -r_{pri} \\ -r_{cent} \end{bmatrix}. \tag{2.30}$$

Then we can form a path depending on the parameter τ as:

$$(x,y,z) + (\triangle x(\tau), \triangle y(\tau), \triangle z(\tau)). \tag{2.31}$$

Algorithm 1 PDIPM Framework

Require: A full rank matrix $A \in \mathbb{R}^{M\times N}$, $M < N$, a vector $b \in \mathbb{R}^M$, initialization $(x^{(0)}, y^{(0)}, z^{(0)})$. Initial slack variable τ and decreasing factor $\xi \in (0,1]$, Iteration $k \leftarrow 0$.

1: Repeat
2: $k \leftarrow k+1$, $\tau \leftarrow \xi\tau$.
3: Solve (2.30) for $(\triangle \mathbf{x}(\tau), \triangle \mathbf{y}(\tau), \triangle \mathbf{z}(\tau))$.
4: Update $\mathbf{x^{(k)}} \leftarrow \mathbf{x^{(k-1)}} + \triangle \mathbf{x}(\tau)$, $\mathbf{y^{(k)}} \leftarrow \mathbf{y^{(k-1)}} + \triangle \mathbf{y}(\tau)$ and $\mathbf{z^{(k)}} \leftarrow \mathbf{z^{(k-1)}} + \triangle \mathbf{z}(\tau)$.
5: **Until** stopping criteria is satisfied.
Output $\mathbf{x}^* \leftarrow \mathbf{x^{(k)}}$.

We can summarize the framework of PDIPM as:

Algorithm 1 requires a total of $O(\sqrt{N})$ iterations, and each iteration can be executed in $O(N^3)$ operations for solving the linear system (2.30). We can also solve the ℓ_1-minimization problem by converting (2.24) into a family of log-barrier problems [EJ06] as:

$$\min_x \; c^T x - \tau \sum_{i=1}^{N} log x_i, \tag{2.32}$$

$$s.t.\; Ax = b,\; x \geq 0.$$

Assuming that the above sets are non-empty, and applying the PDIPM framework we can also solve (2.32). Besides it can be shown that (2.32) has a unique global optimal solution $x(\tau)$ for all $\tau > 0$, and as $\tau \to 0$, $x(\tau, y(\tau), z(\tau))$ converges to the optimal solution of problems $(\mathbf{P})$ and $(\mathbf{D})$ respectively [RI84]. Here we need to point out that PDIPM is computationally expensive mainly because at each iteration, we need to solve a large scale linear system (2.30). For instance, to restore a 1024×1024 image, it is impossible to store the iteration matrix explicitly, and solving such huge linear systems is very expensive.

2.3.2 *Iterative Shrinkage Thresholding Methods and FIST methods*

We point out that, in most applications, e.g. in image deblurring, the sensing matrix $\mathbf{A} = \mathbf{RW}$ is often composed of a wavelet transform $\mathbf{W}$ and a blurring operator $\mathbf{R}$,and it is not only large scale (millions variables) but also involves dense matrix data. Although it is known that the convex program (2.13) can be cast as a **LP** or **SOCP**, it often precludes the use and potential advantage of sophisticated interior point methods because of the high computational cost. Hence, this motivates the search for fast gradient based algorithms for solving (2.13) and algorithms only involving simple operations such as vector algebra and matrix vector multiplications.

The Iterative Shrinkage Thresholding (IST) Methods are well known as fast algorithms [IM04][EY07][AB98][SM08][M.E06] utilizing operator splitting methods. Initially IST was presented as an EM algorithm in the context of image deconvolution problems [MR03]. Generally, the IST method is aimed at solving the problem as:

$$\min_x F(x) = f(x) + \lambda g(x), \tag{2.33}$$

where $f(x) : \mathbb{R}^N \to R$ is a smooth and convex function, the regularization term $g(x) : \mathbb{R}^N \to R$ is bounded below and not necessarily smooth, and λ is the regularization parameter. For the ℓ_1 minimization problem (2.13), the regularization term $g(x)$ is expressed as:

$$g(x) = \sum_i g_i(x_i),$$

where $g_i(x_i) = |x_i|$ and $f(x) = \frac{1}{2}||Ax - b||^2$ is the quadratic term reflecting the noise level. Then the recursion for updating x can be derived by using the second order approximation of the function $f(x)$ as well as a proper approximation of its Hessian

matrix:

$$\begin{aligned} x^{k+1} &= \arg\ \min_x \{f(x^k) + (x-x^k)^T \nabla f(x^k) + \frac{1}{2}(x-x^k)^T \nabla^2 f(x^k)(x-x^k) + \lambda g(x)\} \\ &= \arg\ \min_x \{(x-x^k)^T \nabla f(x^k) + \frac{\alpha^k}{2}||x-x^k||^2 + \lambda g(x)\} \\ &= x^k - \frac{1}{\alpha^k}\nabla f(x^k) - \frac{\lambda}{\alpha^k}\nabla g(x) \\ &\Leftrightarrow \arg\ \min_x \frac{1}{2}||x-\gamma^k||^2 + \lambda g(x), \end{aligned} \tag{2.34}$$

where $\gamma^k = x^k - \frac{1}{\alpha^k}\nabla f(x^k)$, the vector α^k in the second line is an approximation of the diagonal entries in the Hessian matrix $\nabla^2 f(x^k)$. When the separable ℓ_1 regularization term $g(x) = ||x||_1$ is plugged in, x in (2.34) can be processed component wisely via solving each problem:

$$x_i^{k+1} = \arg\ \min_{x_i}\ S_i(x_i) = \frac{1}{2}(x_i - \gamma_i^k)^2 + \frac{\lambda}{\alpha^k}|x_i|, \quad i = 1, \cdots N, \tag{2.35}$$

and each x_i^{k+1} in (2.35) has the closed form solution expressed as:

$$x_i^{k+1} = \begin{cases} \gamma_i^k - \frac{\lambda}{\alpha^k}, & \text{if } \gamma_i^k > \frac{\lambda}{\alpha^k} \\ \gamma_i^k + \frac{\lambda}{\alpha^k}, & \text{if } \gamma_i^k < -\frac{\lambda}{\alpha^k} \\ 0, & \text{otherwise} \end{cases} \tag{2.36}$$

and equivalently, the closed form solution (2.36) can expressed in terms of soft thresholding or shrinkage [D.L95]:

$$x_i^{k+1} = soft(\gamma_i^k, \frac{\lambda}{\alpha^k}) = sgn(\gamma_i^k)\max(|\gamma_i^k| - \frac{\lambda}{\alpha^k}, 0). \tag{2.37}$$

Hence the solution of (2.13) can be obtained component wise. IST methods take advantage of operator splitting and each component in the solution can be processed in parallel, so this structure is especially suited for the large scale problem and much faster than the traditional PDIPM. The parameters α^k, λ in (2.37) play an important role and a sophisticated strategy for determining them is required. Matrix $\alpha_{\mathbf{k}}\mathbf{I}$ is an approximation of the Hessian matrix $\nabla^2 f(x)$ and there are many strategies for updating

it. For instance, the α^k can be defined as the minimizer of the discrepancy between $\alpha_{\mathbf{k}}\mathbf{I}$ and the Hessian matrix as:

$$\begin{aligned}\alpha^{k+1} &= \arg\ \min_{\alpha} ||\alpha\mathbf{I} - \nabla^2 f(x^k)||^2 \\ &\approx \arg\ \min_{\alpha} ||\alpha - \frac{\nabla f(x^k) - \nabla f(x^{k-1})}{x^k - x^{k-1}}||^2 \\ &= \frac{(x^k - x^{k-1})^T(\nabla f(x^k) - \nabla f(x^{k-1}))}{||x^k - x^{k-1}||^2},\end{aligned} \tag{2.38}$$

where $\nabla^2 f(x) \approx \frac{\nabla f(x^k) - \nabla f(x^{k-1})}{x^k - x^{k-1}}$ is the first order approximation of the Hessian obtained by using the values from the previous two steps. This strategy is also referred to as Barzilai Borwein (**BB**) equation [BB88]. The parameter λ in (2.37) is the regularization parameter for balancing the Euclidean norm fidelity term and the ℓ_1 regularization. Then the optimal ℓ_1 regularized solution can be reached as $\lambda \to 0$. In practice, λ can be initialized with a relative large value and be reduced in each step to let it approach to zero gradually, instead of assigning a small value to it and this may degrade the convergence. This so called warm start strategy has been widely used and it is also referred to as continuation [EY07][MS07a]. The algorithm of IST can be summarized as the **Algorithm 2**.

Algorithm 2 IST Framework

Require: A full rank matrix $A \in \mathbb{R}^{M\times N}$, $M < N$, a vector $b \in \mathbb{R}^M$, initialize $x^{(0)}, \alpha^0$, warm start λ, and the decreasing factor $\xi \in (0,1]$, Iteration $k \leftarrow 0$.

1: Repeat
2: $k \leftarrow k+1$, $\lambda \leftarrow \xi\lambda$.
3: x_i^k is updated from (2.37), where $i = 1, \cdots, N$
4: α^k is updated from (2.38)
5: **Until** stopping criteria is satisfied.
Output $\mathbf{x}^* \leftarrow \mathbf{x}^{(\mathbf{k})}$.

It is worth to point out that the IST method possesses simplicity and it only requires the function value and the gradient valuation, so this method belongs to the first order methods which is ideal for the large scale problem in practice, however, the sequence $\{x^k\}$ generated by the IST algorithm may converge quite slowly. On the

one hand, this is mainly because of the limitations of the first order method, on the other hand, the highly dependence on the dense sensing matrix may also lead to slow convergence. Theoretically, the IST algorithm possesses a sub-linear global rate of convergence, and behaves like:

$$F(x^k) - F(x^*) \approx O(\frac{1}{k}). \tag{2.39}$$

Hence this motivates finding a fast iterative soft shrinkage (FIST) [AM09] method that combines the simplicity of IST with a faster global rate of convergence both theoretically and computationally.

A new attempt for accelerating the IST is to use the sequential subspace optimization techniques [AM09][JDM98][MM07][Y.E07] and to generate the new iteration by minimizing a function over an affine subspace spanned by two previous iterations and the current gradient value. A new proposed two-step IST algorithm, namely TwIST [JDM98], shows better convergence results; recently an unpublished work written by Nesterov [Y.E07] reveals that a multi-step version of an accelerated first order method that solves (2.33) is proven to converge in function values as $O(\frac{1}{k^2})$.

Algorithm 3 FIST Framework

Require: Given a full rank matrix $A \in \mathbb{R}^{M\times N}$, $M < N$ and a vector $b \in \mathbb{R}^M$,

1: Set $x^0 = 0, x^1 = 0, t^0 = 1, t^1 = 1$, $k \leftarrow 1$.
2: Initialize λ^0, $\beta \in (0,1)$ and $\bar{\lambda} > 0$.
3: Repeat
4: Update y^{k+1} via (3.9).
5: Update L^{k+1} via (2.44) for given y^{k+1}.
6: Update $x^{k+1} \leftarrow soft(u^k, \frac{\lambda_k}{L_k})$, where $u^k = y^k - \frac{1}{L^k}\nabla f(y^k)$.
7: Update $t^{k+1} = \frac{1+\sqrt{(t^k)^2+1}}{2}$.
8: Update $\lambda^{k+1} \leftarrow max(\beta\lambda^k, \bar{\lambda})$.
9: $k \leftarrow k+1$.
10: **Until** stopping criteria is satisfied.
Output $\mathbf{x}^* \leftarrow \mathbf{x}^{(\mathbf{k})}$.

The principle of the FIST method is that we apply second order expansion of $f(x)$ in (2.33) around a well defined point y that could be defined via a linear combina-

tion of the points obtained in the previous two iterations to approximate $f(x)$. Suppose $\nabla f(x)$ is Lipschitz continuous with Lipschitz constant L, then (2.33) can be approximated as:

$$Q(x,y) \approx f(y) + (x-y)^T \nabla f(y) + \frac{L}{2}||y-x||^2 + \lambda g(x), \tag{2.40}$$

and $F(x) \leqslant Q(x,y)$ for all y. Similar to the derivation in (2.34), the minimizer of (2.40) in terms of y value can be solved via soft thresholding as:

$$\arg \min_x Q(x,y) = \arg \min_x \frac{L}{2}||x-u||^2 + \lambda g(x), \tag{2.41}$$

where $u = y - \frac{1}{L}\nabla f(y)$ Then its closed form solution can be expressed through the soft thresholding as:

$$\arg \min_x Q(x,y) = soft(u, \frac{\lambda}{L}) \tag{2.42}$$

and y can be updated by using the previous two iterations as

$$y^k = x^k + \frac{t^{k-1}-1}{t^k}(x^k - x^{k-1}), \tag{2.43}$$

where $\{t^k\}$ is a positive sequence satisfying $(t^k)^2 - t^k \leq t^{k-1}$ such that the convergence rate reaches $O(\frac{1}{k^2})$ [AM09]. For the large scale problem, a backtracking line search scheme [AM09] can be used to generate the Lipschitz constant sequence $\{L^k\}$ by finding the smallest nonnegative integers i_k such that for $\eta > 1$ with $L^k = \eta^{i_k} L^{k-1}$ the following inequality holds:

$$F(P_{L^k}(y^k)) \leq Q_{L_k}(P_{L^k}(y^k), y^k), \tag{2.44}$$

where $P_{L^k}(y) \triangleq \min_x Q_{L^k}(x,y) = soft(u, \frac{\lambda}{L^k})$ and $u = y - \frac{1}{L_k}\nabla f(x)$. Hence the rate of convergence has been proved and given in [AM09] as:

$$F(x^k) - F(x^*) \leq \frac{2L||x^0 - x^*||^2}{(1+k)^2}. \tag{2.45}$$

The FIST algorithm is summarized in **Algorithm 3**, the proof and detailed theoretical analysis is available in [AM09].

2.3.3 Gradient Projection Method

The Gradient Projection for Sparse Reconstruction (GPSR) [MS07a] is used to detect a sparse representation of the objective along a certain gradient direction, which shows a fast convergence speed and robustness in computation. GPSR solves the ℓ_1-norm regularized linear problem (2.13) by reformulating it into a quadratic programming (QP).

We can start to discuss the GPSR from the equivalent unconstrained problem of (2.13):

$$\min_x \ \frac{1}{2}||Ax-b||^2+\lambda||x||_1, \tag{2.46}$$

where $\lambda > 0$ is the regularization parameter and as $\lambda \to 0$, the solution of (2.46) converges to the optimal solution. First we introduce vectors u and v and make the substitution

$$\mathbf{x} = \mathbf{u}-\mathbf{v}, \ \ \mathbf{u} \geqslant \mathbf{0}, \mathbf{v} \geqslant \mathbf{0}. \tag{2.47}$$

Thus $||\mathbf{x}||_{\mathbf{1}} = \mathbf{1^T u} + \mathbf{1^T v}$, where $\mathbf{1} = [1, \cdots, 1]^T$, and the problem (2.46) can be reformulated as:

$$\begin{aligned} &\min_{u,v} \ \frac{1}{2}||b-A(u-v)||^2+\lambda 1^T u+\lambda 1^T v, \\ &s.t. \ \ u \geqslant 0, \ \ v \geqslant 0. \end{aligned} \tag{2.48}$$

The (2.48) can be rewritten in standard QP form as:

$$\begin{aligned} &\min_z \ \ Q(z) \triangleq c^T z+\frac{1}{2}z^T Bz, \\ &s.t. \ \ z \geqslant 0, \end{aligned} \tag{2.49}$$

where $z = [u,v]^T$, $y = A^T b$, $c = \lambda 1 + [-y,y]^T$ and

$$B = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}. \tag{2.50}$$

The gradient of $Q(z)$ is $\nabla_z Q(z) = c + Bz$, and the variable z^k is updated in each iteration by a steepest descent algorithm which moves along the negative gradient direction $-\nabla_z Q(z)$ with a certain step length α^k from each iteration z^k as:

$$z^{k+1} = z^k - \alpha^k \nabla Q(z^k). \tag{2.51}$$

Specifically, we define the vector g^k by

Algorithm 4 GPSR Framework

Require: Given the full rank matrix $A \in \mathbb{R}^{M\times N}$, $M < N$, a vector $b \in \mathbb{R}^M$ and $z^{(0)}, \alpha^0$, warm start λ, and the decreasing factor $\xi \in (0,1]$, Iteration $k \leftarrow 0$.

1: Repeat
2: $k \leftarrow k+1$, $\lambda \leftarrow \xi\lambda$.
3: Update α^k via backtracking (2.53)
4: z^k is updated from (2.51), where $i = 1, \cdots, N$
5: **Until** stopping criteria is satisfied.
Output $\mathbf{x}^* \leftarrow \mathbf{x}^{(\mathbf{k})}$.

$$g_i^k = \begin{cases} (\nabla Q(z^k))_i, & \text{if } z_i^k < 0 \text{ or } \nabla Q(z^k))_i < 0, \\ 0, & \text{otherwise} \end{cases} \tag{2.52}$$

The step length α^k can be determined by $\alpha^k = \arg\ \min_\alpha\ F(z^k - \alpha g^k)$ which can be computed explicitly as

$$\alpha^k = \frac{(g^k)^T g^k}{(g^k)^T B g^k}. \tag{2.53}$$

The GPSR framework can be summarized in **Algorithm 4** with a warm start scheme on λ. We need to point out that the dimension of the problem is doubled when we convert the problem into a QP and the computational complexity and rate of convergence still have no explicit estimation [MS07a].

In this section, we first reviewed the recently developd theory of compressive sensing as well as some of its applications; secondly several current widely used efficient sparse optimization algorithms are discussed. In next section, we will propose our algorithm, which is based on a complete different method with efficient accelerating schemes, for solving the large scale sparse optimization especially for the compres-

sive sensing applications; an in-depth discussion on the convergence properties of the proposed algorithm are covered.

Chapter 3

The TV and ℓ_1-norm Regularized Sparsity Enforcing Algorithm

In this chapter, we present a decomposition algorithm for solving the convex programming problem which can be extended to solve the TV and ℓ_1 regularized inverse problem for restoring the sparsity features. The proposed algorithm is motivated by the wide range of applications of the sparse optimization techniques supported by the recent developed theory of compressive sensing [E.C06][E.C04][E.C05][D.L06][DA92] as well as the relative computational issues encountered when the large scale data set and dense matrix are involved. The proposed algorithm is based on an augmented Lagrangian mutiplier methods. By taking the advantage of separable structure in the objective function, the proposed algorithm is fit for large scale problems and has a parallel processing feature. Besides, under the assumptions that both the primal and dual problems have at least one solution and the solution of the primal problem is bounded, the global convergence of the algorithm is established.

Decomposition of problems is an efficient way to process the large scale problem and these methods attract the interest of researchers. As in the context of image processing, the sparse optimization problems arising in compressive sensing and wavelet imaging are naturally large scale. Since the block structure and the sharp jumps in the images need to be restored precisely, and since large dense matrices and ℓ_1-norm are involved, the traditional interior point methods and some current first order methods may encounter the problems of large matrix storage, high computational load and slow convergence. Therefore, it is necessary to develop a new algorithm that is satisfied for the natural of ℓ_1-norm related sparse optimization and its relative applications in wavelet imaging and compressive sensing.

The proposed algorithm is based on an augmented Lagrangian framework for solving a TV and ℓ_1-norm regularized inverse problem. In this method, we first refor-

mulate the objective as an unconstrainted problem by using the augmented Lagrangian method. Next we update the primal variables by taking advantage of the separable structure in the TV and ℓ_1-norm objective. We introduce slack variables to split the problem into several blocks where each block involves the sum of a component of primal variables or its functional and a quadratic discrepancy between the primal variables and the slack variable. Then the augmented Lagrangian function is minimized by solving the primal variables with fixed slack variables and doing this alternatively in the inner iteration to update the slack variables. The whole computation for updating each component of the slack variable only requires some simple scalar products and can be processed in parallel, and the primal variables are calculated via preconditioned conjugate gradients taking advantage of the block Toeplitz iteration matrices. Finally the dual variable is updated by solving the dual problem with fixed primal variables, and the augmented Lagrangian multiplier is updated at outer iteration to accelerate the convergence. This gives us the multi splitting augmented Lagrangian method.

Our contributions are as follows. we present a fast algorithm for solving the TV and ℓ_1-norm regularized inverse problem and the implementation is tested in MR imaging application with clinic data. We compare our method with some current sparse optimization methods to show that our method is generally comparable with other packages and in some sides our method shows better numerical performance. Moreover, some simple proof for the global convergence and convergence speed is shown. We also construct a preconditioned for the blocky Toeplitz matrices and finally we test it with numerical experiments.

This chapter is organized as follows. We first present the general framework of the proposed algorithm based on the augmented Lagrangian methods and then we prove its global convergence and discuss the calibration of parameters in the algorithms. Next we extend our discussion onto the practical issues related to the design of the preconditioner and how the regularization parameters affect the rate of convergence.

Some theoretical and experimental analysis is presented to demonstrate the robustness and efficiency of our algorithms.

3.1 A multi-splitting method based on augmented Lagrangian function for separable convex problem

Consider a separable convex problem:

$$\begin{aligned} &\min_{x\in\mathbb{R}^N} \quad \sum_i f_i(x_i) \\ &s.t. \;\; \mathscr{A}(x) = b, \\ &x_i \in x \subset \mathbb{R}^N, \;\; i = 1,\cdots N, \end{aligned} \tag{3.1}$$

where $f_i : \mathbb{R} \to \mathbb{R}$ are convex functions, x is nonempty closed convex subsets of $\mathbb{R}^N$, the linear map $\mathscr{A}(\cdot) : \mathbb{R}^N \to \mathbb{R}^M$ is defined as:

$$\mathscr{A}(x) := \sum_i A^{(i)} x_i, \;\; i = 1,\cdots,N, \tag{3.2}$$

the matrix A and the vector $b \in \mathbb{R}^M$ are given, $x \in \mathbb{R}^N$ is the unknown vector. Note that equation $\mathscr{A}(x) = b$ is equivalent to $Ax = b$, where A is defined as:

$$A := [A^{(1)},\cdots,A^{(N)}] \in \mathbb{R}^{M\times N}.$$

We make the following assumption throughout our presentation in this work:

Assumption 3.1.1. *The matrix A is of full row rank and the Slater's condition is satisfied for (3.1), that is there exist a vector $\tilde{x}$ such that $\mathscr{A}(\tilde{x}) = b$.*

Let $y \in \mathbb{R}^M$ be a vector of Lagrangian multipliers. The augmented Lagrangian function of the primal problem $L_r : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}$ is given by:

$$L_r(x,y) = \sum_i f_i(x_i) + y^T(Ax-b) + \frac{r}{2}||Ax-b||^2, \tag{3.3}$$

where $r > 0$ is a positive parameter for penalizing the additional quadratic term. We note that by adding an additional quadratic term to the traditional Lagrangian function,

the so-called augmented Lagrangian function (3.3) becomes a strictly convex function and this improves the convergence of the algorithm.

Starting from the initial value of the dual variable y^0, the augmented Lagrangian method solves in the k-th iteration

$$\min_{x\in\mathbb{R}^N} \; L_r(x,y^k), \tag{3.4}$$

for x^{k+1}, and then updates the dual variable y^{k+1} by

$$y^{k+1} = y^k + \rho(Ax^{k+1} - b). \tag{3.5}$$

Since solving the problem (3.4) is very expensive, by taking advantage of the separable structure of the objective function, we introduce slack variables into (3.4) such that it can be separated into several subproblems. At the k-th iteration the augmented Lagrangian function (3.4) can be reformulated into:

$$\min_{x,w\in\mathbb{R}^N} \; L_r(x,w,y^k) = \sum_i w_i + \frac{r}{2}\sum_i (w_i - f_i(x_i))^2 + y^k(Ax-b) + \frac{r}{2}||Ax-b||^2, \tag{3.6}$$

where $w \in \mathbb{R}^N$ is the slack variable. Then starting from the initial value of the primal and dual variables x^0 and y^0, at the k-th iteration we update w,x and y by first minimizing $L_r(x,w,y)$ with respect to w to obtain w^{k+1} with $x = x^k$ and $y = y^k$ fixed; then minimize $L_r(x,w,y)$ with respect to x to obtain x^{k+1} with $w = w^{k+1}$ and $y = y^k$ fixed; and finally the dual variable $y = y^{k+1}$ is updated via (3.5) with $x = x^{k+1}$ fixed. Hence we can express these as:

$$w^{k+1} = \arg\min_{w} \; L_r(x^k,w,y^k) \tag{3.7}$$

$$x^{k+1} = \arg\min_{x} \; L_r(x,w^{k+1},y^k) \tag{3.8}$$

$$y^{k+1} = y^k + \rho(Ax^{k+1} - b). \tag{3.9}$$

So far we can summarize the proposed general framework of the proposed multi-splitting method for solving a separable objective function with linear constraint as:

Algorithm 5 The General Framework of the Proposed Multi-Splitting Method Based on Augmented Lagrange

1: Set x^0 and $y^0 > 0$.
2: **for** $k = 0, 1, \cdots$ **do**
3: Compute w^{k+1} according to (3.7).
4: Compute x^{k+1} according to (3.8).
5: Compute y^{k+1} according to (3.9).
6: **end for**

The **Algorithm 5** is motivated by the problems raising in image processing and now we extend it to solve the TV and ℓ_1 regularized inverse problem:

$$\begin{aligned} \min_{u} \quad & \alpha||\Psi^T u||_{\ell_1} + \beta TV(u) \\ s.t. \quad & Au = b, \end{aligned} \tag{3.10}$$

where $u \in \mathbb{R}^N$ is obtained by vectoring the pixels in a $\sqrt{N}$ square image along each column, Ψ is an orthogonal sparse transformation such that $u = \Psi a = \sum_i \Psi_i a_i$ has a sparse representation under it. α and β are positive weight coefficients of the relative TV and ℓ_1 terms. A partial observation $b \in \mathbb{R}^M$ is obtained via a sensing matrix $A \in \mathbb{R}^{M\times N}$ ($M \ll N$). In practice, the sensing matrix A has various versions, such as partial Fourier, partial DCT *ect*., which depends on the specific problem and it should also meet the RIP conditions in the context of compressive sensing. We need to point out that the objective function in (3.10) possesses a separable structure, such as:

$$TV(u) = \sum_i ||Du_i||, \tag{3.11}$$

$$||\Psi^T u||_{\ell_1} = \sum_i |\Psi_i^T u_i|, \tag{3.12}$$

where $||\cdot||$ denotes the Euclidean norm, $D = [D_1; D_2]$ is the finite difference operator to the i-th pixel in u, D_1, D_2 represents the relative row and column difference operator, and Ψ_i^T is the i-th column in the sparse transformation Ψ. Hence the problem (3.10)

can be expressed as:

$$\begin{aligned} \min_{u} \quad & \alpha \sum_{i}^{N} h_i(u_i) + \beta \sum_{i}^{N} g_i(u_i) \\ s.t. \quad & Au = b, \end{aligned} \tag{3.13}$$

where $h_i(u_i) = |\Psi_i^T u_i|$ and $g_i(u_i) = ||Du_i||$. Then we can adapt the **Algorithm 5** to solve model (3.13) by the augmented Lagrangian function based multi-splitting method as follows. We start from the augmented Lagrangian function of (3.13) and write it as:

$$L_r(u,y) = \alpha \sum_{i}^{N} h_i(u_i) + \beta \sum_{i}^{N} g_i(u_i) + y^T(Au-b) + \frac{r}{2}||Au-b||^2. \tag{3.14}$$

Given the initial value of u^0 and y^0, at the k-th iteration $u^{k+1} = \arg\ \min_u L_r(u, y^k)$ is updated with fixed y^k, and similarly we can split this problem into several subproblems by introducing the slack variables w and v as follows:

$$L_r(u,w,v,y^k) = \sum_{i} G_i(u_i,w_i,v_i) + {y^k}^T(Au-b) + \frac{r}{2}||Au-b||^2, \tag{3.15}$$

where $G_i(u_i,w_i,v_i) = \alpha w_i + \frac{r}{2}(w_i - h_i(u_i))^2 + \beta v_i + \frac{r}{2}(v_i - g_i(u_i))^2$. Hence starting from the initial u^0 and y^0, the variables u, w, v, y are updated as follows:

$$w_i^{k+1} = \arg\min_{w_i} L_r(u_i^k, w_i, v_i, y^k), \tag{3.16}$$

$$v_i^{k+1} = \arg\min_{v_i} L_r(u_i^k, w_i, v_i, y^k), \tag{3.17}$$

$$u^{k+1} = \arg\min_{u} L_r(u, w^{k+1}, v^{k+1}, y^k), \tag{3.18}$$

$$y^{k+1} = y^k + \rho(Au^{k+1} - b) \tag{3.19}$$

The subproblems (3.16) and (3.17) are processed component-wise and the closed form solutions are expressed via soft thresholding as:

$$\begin{aligned} w_i^{k+1} &= soft(|\Psi^T u_i^k|, \frac{\alpha}{r}) \triangleq max\{|\Psi^T u_i^k| - \frac{\alpha}{r}, 0\} sgn(\Psi^T u_i^k), \\ v_i^{k+1} &= soft(||Du_i^k||, \frac{\beta}{r}) \triangleq max\{||Du_i^k|| - \frac{\beta}{r}, 0\} sgn(Du_i^k), \end{aligned}$$

and in this way the slack variable w, v play an important role in splitting the large scale problem into simple subproblems that have the closed form solutions updated via simple scalar multiplication in terms of soft-thresholding. The subproblem (3.18) reduces to a quadratic problem with positive definite block Toeplitz Hessian matrix and it can be solved quickly via the preconditioned conjugate gradient method which will be discussed in the following sections. Moreover, the regularization parameter r in augmented Lagrangian formula (3.3) balances the fidelity and the regularization terms, an appropriate choice on its value can accelerated the convergence of the routine. A common and efficient scheme is a so call warm start strategy for updating r, that is initially we assign a relative small value to it and update it in the outer-loop till the solution converges along a path of r. We summarize the algorithm of multi-splitting method for solving the TV and ℓ_1 regularized problem as follow:

Algorithm 6 Multi-Splitting Methods for TV-ℓ_1 Regularized Inverse Problem

Require: $A, b, u^0, y^0, \alpha, \beta, r > 0, \rho > 0$

for Outterloop = $0, 1, \cdots$ **do**
  Set u^0 and $y^0 > 0$.
  for $k = 0, 1, \cdots$ **do**
    Compute w^{k+1} according to (3.16).
    Compute v^{k+1} according to (3.17).
    Compute u^{k+1} according to (3.18).
    Compute y^{k+1} according to (3.19).
  end for
  Update r.
end for

3.2 The optimality and convergence analysis

In this section, we present some theoretical analysis on the proposed algorithms. We will show the optimality conditions and prove the convergence of the proposed algorithm. Moreover we will discuss the optimal choice of the parameters and show how their values affect the convergence rate.

3.2.1 The optimality

The **Algorithm 5** is based on the general augmented Lagrangian function, where the primal variables are updated by fixing the dual variables obtained in previous iteration and in practice, this step can be decomposed into several simple subproblems with closed form solutions by taking the advantage of the separable objective. Finally the dual variable is updated by using the updated primal variables. Since the problem (3.1) has a convex objective with linear constraint and the augmented Lagrangian with an additional quadratic term is a strict convex function, the **KKT** condition becomes a necessary and sufficient condition on the optimal solution when the **Assumption 3.1.1** is satisfied.

Theorem 3.2.1. *Suppose (3.1) has a nonempty and bounded solution set, and Slater's condition is satisfied, that is the feasiable solution exits. Then the sequence* $\{x^k\}$ *generated via (3.7)-(3.9) in* **Algorithm 5** *is bounded and every limit point* $\lim_{k\to\infty} x^k = x^*$ *is the solution of problem (3.1).*

Proof. Given $x^0, y^0 > 0, r > 0$, and suppose the sequence $\{x^k, y^k, w^k\}$ generated via **Algorithm 5** has a unique limit point $\{x^*, y^*, w^*\}$ as $k \to \infty$. Then from (3.9) we have:

$$y^* = y^* + \rho(Ax^* - b) \Leftrightarrow Ax^* = b, \tag{3.20}$$

and from (3.7), we have $w_i^{k+1} = f_i(x_i^k) - \frac{1}{r}$. Since x^* minimizes (3.8) and the solution set is nonempty, we have

$$\begin{aligned} &0 \in \partial\, L_r(x, w^{k+1}, y^k) \\ &\Leftrightarrow 0 \in -diag(r(f_i(x_i^k) - \frac{1}{r} - f_i(x^*)))\partial f(x^*) + A^T y^k + rA^T(Ax^* - b), \end{aligned}$$

and this implies that

$$\Rightarrow \partial f(x^*) + A^T y^k = 0, \;\; as\; k \to \infty, \tag{3.21}$$

where $f(x) = (f_1(x_1), \cdots, f_N(x_N))^T$. Hence from (3.20) and (3.21), the primal-dual pair $\{x^*, y^*\}$ satisfies the **KKT** condition of (3.1). □

3.2.2 The convergence proof

As in the context of the TV and ℓ_1 regularized inverse problem (3.10), we are interested in the convergence of the proposed splitting algorithm. In (3.19), we have the freedom of the choice of the parameter ρ when updating the dual variable y. The ρ value plays an important role in determining the rate of convergence and an optimal choice of its value is determined via studying the dual problem. We will present the convergence of **Algorithm 6** and study how parameter values affect its convergence rate.

Suppose the optimal primal dual pair $\{u^*, y^*\}$ in (3.14) is a saddle point of the augmented Lagrangian function $L_r(u, y)$, that is at the k-th iteration if the current value is given as $\{u^k, y^k\}$, then its value should satisfy:

$$L_r(u^*, y^k) \leq L_r(u^*, y^*) \leq L_r(u^k, y^*). \tag{3.22}$$

Therefore the primal dual pair (u^k, y^k) generated via **Algorithm 6** is characterized by

$$\begin{cases} L_r(u^{k+1}, y^k) \leq L_r(u^k, y^k) \\ y^{k+1} = y^k + \rho(Au^{k+1} - b) \end{cases} \tag{3.23}$$

Then for the fixed slack variable values w and v, the saddle point (x^*, y^*) satisfies

$$(I + D^T D + A^T A)u^* + \frac{1}{r}A^T y^* = \Psi w + D^T v + A^T b, \tag{3.24}$$

$$Au^* = b \;\Leftrightarrow\; y^* = y^* + \rho(Au^* - b), \tag{3.25}$$

where (3.24) is equivalent to the normal equations derived from (3.18).

Theorem 3.2.2. *For all $y^0 \in \mathbb{R}^M$ and $u^0 \in \mathbb{R}^N$, the sequence $\{u^k\}$ generated via* **Algorithm 6** *converges to the solution of (3.10) u^* if and only if $0 < \rho \leqslant 2r$ as $k \to \infty$.*

Proof. Let us define the difference as $\hat{u}^k \triangleq u^k - u^*$ and $\hat{y}^k \triangleq y^k - y^*$.

Then we have $\hat{y}^{k+1} = \hat{y}^k + \rho A\hat{u}^{k+1}$, by squaring this quantity and rearranging $\hat{y}^k$ to left side, we have:

$$(\hat{y}^{k+1})^2 - (\hat{y}^k)^2 = 2\rho < A\hat{u}^{k+1}, \hat{y}^k > +\rho^2 < A\hat{u}^{k+1}, A\hat{u}^{k+1} > . \tag{3.26}$$

From equation (3.22), we have

$$(I + D^T D + A^T A)\hat{u}^{k+1} + \frac{1}{r}A^T\hat{y}^k = 0. \tag{3.27}$$

We multiply by $\hat{u}^{k+1}$ on both sides of (3.27) and rewrite it as:

$$< A^T A\hat{u}^{k+1}, \hat{u}^{k+1} >= - < (I + D^T D)\hat{u}^{k+1}, \hat{u}^{k+1} > -\frac{1}{r} < A^T\hat{y}^k, \hat{u}^{k+1} > \tag{3.28}$$

By plugging (3.26) into (3.28) we have:

$$\Rightarrow (\hat{y}^{k+1})^2 - (\hat{y}^k)^2 = \rho(2 - \frac{\rho}{r}) < A\hat{u}^{k+1}, \hat{y}^k > -\rho^2 < (I + D^T D)\hat{u}^{k+1}, \hat{u}^{k+1} > .$$

It follows from (3.27) that

$$< A\hat{u}^{k+1}, \hat{y}^k >= -r < (I + D^T D + A^T A)\hat{u}^{k+1}, \hat{u}^{k+1} > .$$

Then we have

$$\begin{aligned}(\hat{y}^{k+1})^2 - (\hat{y}^k)^2 =& -2r\rho < (I + D^T D)\hat{u}^{k+1}, \hat{u}^{k+1} > \\ & -\rho(2r - \rho) < A^T A\hat{u}^{k+1}, \hat{u}^{k+1} > .\end{aligned} \tag{3.29}$$

Hence for any $0 < \rho \leqslant 2r$, the sequence $\{\hat{y}^k\}$ decreases and is bounded below by zero because matrix $A^T A$ is positive definite, and the right hand side implies that $\hat{u}^{k+1} \to 0$ as $k \to \infty$. □

Comments: $0 < \rho \leqslant 2r$ is the necessary condition for the convergence of **Algorithm** 6, and when the sensing matrix A is orthonormal in the compressive sensing application, the convergence condition in *Theorem* 3.2.2 on ρ can be relaxed as $0 < \rho \leqslant 2r(4 + \gamma_i)$

where γ_i represents the i-th eigenvalue of matrix D^TD. When $A^TA = I$, (3.29) can be rewritten as:

$$\begin{aligned}(\hat{y}^{k+1})^2 - (\hat{y}^k)^2 &= -\rho(\hat{u}^{k+1})^T\{2r(I+D^TD) + (2r-\rho)I\}\hat{u}^{k+1} \\ &= -\rho(\hat{u}^{k+1})^T\{(4r-\rho)I + 2rD^TD\}\hat{u}^{k+1}.\end{aligned} \tag{3.30}$$

Then the system $(4r-\rho)I + 2rD^TD$ is positive definite when $4r - \rho + 2r\gamma_i \geqslant 0$, that is $\rho \leqslant 2r(2+\gamma_i)$.

By the **Theorem 3.2.2** we prove the global convergence of the proposed algorithm and we found that the convergence is satisfied under a general condition. Next we will study the rate of convergence and show how the optimal choice of ρ is achieved.

3.2.3 The convergence rate and optimal parameter selection

The dual problem of (3.14) is

$$g(y) = \inf_u \; L_r(u,y),$$

and from (3.22), the optimal primal variable x^* can be expressed in terms of the dual variable y for fixed w and v values as:

$$x^* = (I + D^TD + A^TA)^{-1}(\Psi w + D^Tv + A^Tb - \frac{1}{r}A^Ty). \tag{3.31}$$

Then the dual variable can be solved from the problem given as below:

$$y = \arg \max_y \inf_x L_r(x,y), \tag{3.32}$$

or equivalently solve y from the linear equation determined by the problem $\min_y -L_r(x^*,y)$ as:

$$\frac{1}{r}A(I + D^TD + A^TA)^{-1}A^Ty = A(I + D^TD + A^TA)^{-1}(\Psi w + D^Tv + A^Tb) - b. \tag{3.33}$$

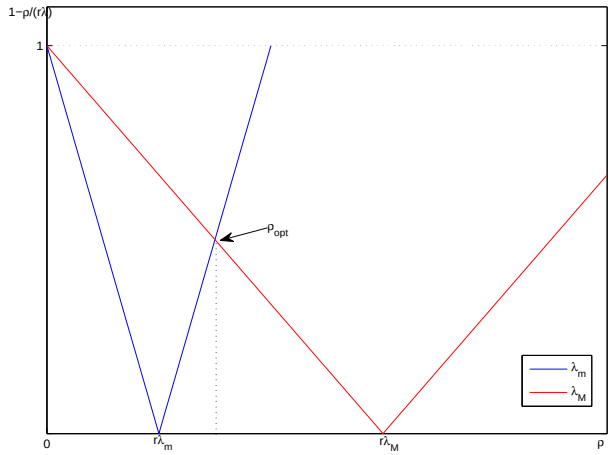

Figure 3.1: The optimal choice of ρ.

More precisely, by eliminating u^k from (3.9) in the proposed algorithm and updating the dual variable y^k via:

$$\begin{aligned} y^{k+1} = y^k &- \frac{\rho}{r}\mathscr{H}^{-1}A^T y^k \\ &- \rho(A\mathscr{H}^{-1}(\Psi w + D^T v + A^T b) + b), \end{aligned} \tag{3.34}$$

where

$$\mathscr{H} \triangleq I + D^T D + A^T A. \tag{3.35}$$

The formula (3.34) derived from the dual problem is only used for theoretical analysis, but in practice, the advantage of the formula (3.9) comparing with (3.34) is that it does not depend on the explicit expression of the inverse matrix of $\mathscr{H}$ and in some practical application it is impossible to calculate $\mathscr{H}^{-1}$ since it is usually large scale.

Let $\hat{y}^k \triangleq y^k - y^*$ where y^* stands for the optimal dual variable, then from (3.34) we have:

$$\hat{y}^{k+1} = \hat{y}^k - \frac{\rho}{r}A\mathscr{H}^{-1}A^T\hat{y}^k = (I - \frac{\rho}{r}A\mathscr{H}^{-1}A^T)\hat{y}^k, \tag{3.36}$$

and if we multiply A^T on both sides (3.36) becomes:

$$\begin{aligned} A^T\hat{y}^{k+1} &= A^T(I-\frac{\rho}{r}A\mathscr{H}^{-1}A^T)\hat{y}^k \\ &= (I-\frac{\rho}{r}A^TA\mathscr{H}^{-1})A^T\hat{y}^k. \end{aligned} \tag{3.37}$$

Denote $\bar{Y}^{k+1} \triangleq A^T\hat{y}^{k+1}$ and rewrite the above formula as

$$\bar{Y}^{k+1} = (I-\frac{\rho}{r}(\mathscr{H}(A^TA)^{-1})^{-1})\bar{Y}^k, \tag{3.38}$$

where the sequence $\{\bar{Y}^k\}_{k\geqslant 0}$ plays an important role in proving the linear convergence of the dual variable. The convergence of the primal variable $\{u^k\}$ is also linear related to $\{\bar{Y}^k\}_{k\geqslant 0}$. Next we are trying to express the rate of convergence in terms of the eigenvalues of matrix $\mathscr{H}(A^TA)^{-1}$ and study the behavior of ρ to show how it affects the convergence rate such that a optimal choice of it can be deduced.

Let λ_i denote the eigenvalue of $\mathscr{H}(A^TA)^{-1}$, and

$$\lambda_m = \lambda_1 \leqslant \lambda_2 \leqslant \cdots\cdots \leqslant \lambda_N = \lambda_M,$$

where λ_M and λ_m denote its max and min eigenvalues respectively. Then according to the eigenvalue decomposition, there exist the orthonormal matrix H such that the symmetric matrix can be expressed as:

$$\mathscr{H}(A^TA)^{-1} = H^T\Lambda H,$$

where $\Lambda \triangleq diag(\lambda_1,\cdots,\lambda_N)$, then the Euclidean norm of $||\bar{Y}_i^{k+1}||$ in (3.38) can be written in terms of eigenvalues of $\mathscr{H}(A^TA)^{-1}$ and satisfies that:

$$||\bar{Y}_i^{k+1}|| \leqslant \Lambda_i(\rho)||\bar{Y}_i^k||, \;\; i=1,\cdots,N, \tag{3.39}$$

where $\Lambda_i(\rho) \triangleq |(1-\frac{\rho}{r\lambda_i})|$. Figure (3.1) is a plot of $\Lambda_i(\rho)$ as a function of ρ, and according to this plot, we find that its max and min x-intercept is reached at $\rho = r\lambda_M$ and $\rho = r\lambda_m$, and the optimal choice of ρ should be reached at a point that the value

of the function $\Lambda_r(\rho)$ w.r.t. each λ_i should not be increased. Then from the graph the optimal point ρ_{opt} is reached when

$$\frac{\rho_{opt}}{r\lambda_m} - 1 = 1 - \frac{\rho_{opt}}{r\lambda_M}, \tag{3.40}$$

and ρ_{opt} is solved from (3.40) as:

$$\rho_{opt} = 2r\frac{\lambda_m\lambda_M}{\lambda_M + \lambda_m},$$

and combining the result of **Theorem 3.2.2** on $0 < \rho \leqslant 2r$, the optimal choice of ρ is expressed as:

$$\rho_{opt} = \begin{cases} 2r\frac{\lambda_m\lambda_M}{\lambda_M+\lambda_m}, & \text{if } \frac{\lambda_m\lambda_M}{\lambda_M+\lambda_m} \leqslant 1 \\ 2r, & \text{if } \frac{\lambda_m\lambda_M}{\lambda_M+\lambda_m} > 1 \end{cases}$$

and the optimal linear convergence rate is given as:

$$||\bar{Y}^{k+1}|| \leq \Lambda(\rho_{opt})||\bar{Y}^k||.$$

It is worth noting that that when A is a tight frame, that is $A^TA = I$, we can simply have a similar result on the rate of convergence based on the spectral distribution of matrix $\mathscr{H}$, which is important in the compressive sensing application.

3.2.4 The stopping criteria

The stopping criteria is not mentioned in **Algorithm 6**. Although we proved the global convergence of the framework, a wisely designed stopping criteria can accelerate the convergence and enhance the accuracy. But it is difficult to make the decision about when an approximate solution is of sufficiently high precision to terminate the routine. We wish the the approximated solution u to be reasonably close to the optimal one while avoid the expensive computational load involved in finding an overly accurate solution. In general, the proposed algorithm is of first order and possess a simple structure and a global convergence properties. But its convergence speed maybe slow as compared to others.

In problem (3.10), after the slack variables w and v are introduced, it is separated into several subproblems (3.16)-(3.19) and the solution of each subproblem w, v and u is derived via taking the sub-differentiation [Roc70] as:

$$\frac{v}{||v||} + r(v - Du) = 0, \tag{3.41}$$

$$\frac{sgn(w)}{||w||} + r(w - \Psi^T u) = 0, \tag{3.42}$$

$$D^T(Du - v) + (u - \Psi w) + \frac{1}{r}A^T y + A^T(Au - b) = 0. \tag{3.43}$$

The decision of the stopping criteria suggested in [YY08] is motivated by evaluating the sub-differentiation of (3.41)-(3.43) in each iterations. At k-th iteration as:

$$\tau_1 \triangleq \frac{v^k}{||v^k||} + r(v^k - Du^k), \tag{3.44}$$

$$\tau_2 \triangleq \frac{sgn(w^k)}{||w^k||} + r(w^k - \Psi^T u^k), \tag{3.45}$$

$$\tau_3 \triangleq D^T(Du^{k+1} - v^k) + (u^{k+1} - \Psi w^k) + \frac{1}{r}A^T y^k + A^T(Au^{k+1} - b). \tag{3.46}$$

Then the routine is terminated when $\tau \triangleq \max\{||\tau_1||, ||\tau_2||, ||\tau_3||\} \leq tol$, where $tol > 0$ is a user decided value.

However evaluating (3.44)-(3.46) is expensive, and actually we can simply calculate the discrepancy of the variable v, w and u in each iteration, and decide to terminate the routine when the decease becomes not striking. Specifically, we set $\tau_1^k = \frac{||v^k - v^{k-1}||}{||v^k||}$, $\tau_2^k = \frac{||w^k - w^{k-1}||}{||w^k||}$ and $\tau_3^k = \frac{||u^k - u^{k-1}||}{||u^k||}$ and terminate the routine when $\tau^k \leqslant tol$, where $\tau^k \triangleq \max\{\tau_1^k, \tau_2^k, \tau_3^k\}$ and we set $tol = 1e-2$ in practice. This scheme works well for the cases we test. Besides the warm start [EY07][MS07a] strategy can be set up as assign a relative small initial r_0 and a cap $\bar{r}$ such that let r approach to $\bar{r}$ gradually till converge. Now we can revise the **Algorithm 6** by adding the stopping criteria and the warm start strategy. This is summarized in **Algorithm 7**.

Algorithm 7 Multi-Splitting Methods for TV-ℓ_1 Regularized Inverse Problem

Require: $A, b, u^0, y^0, \alpha, \beta, r > 0, \rho > 0, tol > 0$

Set $k \leftarrow 0$,
for $r = r_0 < r_1 \cdots < \bar{r}$ **do**
  while "not converge" **do**
    Compute w^{k+1} according to (3.16).
    Compute v^{k+1} according to (3.17).
    Compute u^{k+1} according to (3.18).
    Compute y^{k+1} according to (3.19).
    if $\tau \leq tol$ **then**
      Return u^{k+1}
    else
      Set $k \leftarrow k+1$
    end if
  end while
end for

3.3 Preconditioning for ill-conditioned BTTB matrices

The major computation in **Algorithm 7** is in updating u via problem (3.18), which is equivalent to solving for u at the k-th iteration from the problem given as below:

$$\Phi(u) \triangleq \min_{u \in \mathbb{R}^N} ||\Psi^T u - w^{k+1}||_2^2 + ||Du - v^{k+1}||_2^2 + ||Au - b||_2^2 + \frac{1}{r}(Au-b)^T y^k. \quad (3.47)$$

Solving the quadratic problem (3.47) is equivalent to solving for u from its normal equation given as:

$$\mathbf{Tu} - \mathbf{f} = \mathbf{0}, \quad (3.48)$$

where the matrix $\mathbf{T} \in \mathbf{R}^{N \times N}$ and vector $\mathbf{f} \in \mathbf{R}^N$ are defined as:

$$\mathbf{T} = D^T D + \Psi\Psi^T + A^T A, \quad (3.49)$$

$$\mathbf{f} = \Psi v + D^T w + A^T b - \frac{1}{r} A^T y^k. \quad (3.50)$$

Here we need to point out that in (3.47), the matrix T is a Hermitian matrix and composed of the sum of three parts, where $D^T D = {D^{(1)}}^T D^{(1)} + {D^{(2)}}^T D^{(2)}$, Ψ is an orthogonal transformation and $\Psi\Psi^T = \mathbf{I}$ and the sensing matrix A is a tight frame in the

applications of our interest. Hence the matrix $\mathbf{T}$ is a block Toeplitz with Toeplitz block (BTTB) matrix.

We use the conjugate gradient (CG) method to solve (3.48) for u, then a well designed preconditioner is required since the distribution of the eigenvalues of $\mathbf{T}$ trend to distributed equally and this structure makes the CG method converges slowly and we prefer a clustered distribution of the eigenvalues. T. Chan [T.F88] proposed a specific circulant preconditioner called the optimal circulant preconditioner which works well for solving the Toeplitz systems. The T. Chan's optimal circulant preconditioner $c_F(T_n)$ for a general Toeplitz matrix T_n as shown in (3.51) is defined as:

$$\min_{W_n \in \Im_F} ||T_n - W_n||_{Fro}, \tag{3.51}$$

where $||\cdot||_{Fro}$ is the Frobenius norm, $\Im_F \triangleq \{F^* \Lambda_n F |\ \Lambda_n$ is any $n \times n$ diagonal matrix$\}$ denotes a collection of all circulant matrices where $F_{j,k} = \frac{1}{\sqrt{n}} e^{\frac{2\pi i j k}{n}}$, $i \equiv \sqrt{-1}$ is a Fourier matrix. Suppose a Toeplitz matrix is defined as:

$$\mathbf{T_n} = \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & \cdots & t_{2-n} \\ \vdots & t_1 & t_0 & \ddots & \vdots \\ t_{n-2} & \cdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & t_{n-2} & \cdots & t_1 & t_0 \end{pmatrix}, \tag{3.52}$$

then the diagonal entry c_k in T. Chan's optimal circulant preconditioner $c_F(T_n)$ are given by [T.F88] as:

$$c_k = \begin{cases} \dfrac{(n-k)t_k + k t_{k-n}}{n}, & 0 \le k \le n-1 \\ c_{n+k}, & 0 < -k \le n-1, \end{cases} \tag{3.53}$$

or equivalently express this process as:

$$c_F(T_n) = F^T \delta(F T_n F^T) F, \tag{3.54}$$

where the operator $\delta(A)$ denotes the diagonal matrix whose diagonal entries are the diagonal of the matrix A.

As for our case, we are interested in finding a preconditioner for the BTTB system:

$$\mathbf{T_{mn}u} = \mathbf{f}, \tag{3.55}$$

where $\mathbf{T_{mn}}$ is partitioned into m blocks along its column and each block T_l for $|l| \leq m-1$ is a n square Toeplitz matrix such as (3.52). A natural choice of the preconditioner for BTTB matrix $\mathbf{T_{mn}}$ should be $c_F(T_{mn})$, which is obtained by applying the T. Chan's optimal circulant preconditioner showed in (3.54) to each Toeplitz block in $\mathbf{T_{mn}}$. Mathematically, this process is equivalent to firstly applying the two FFTs to each block in $\mathbf{T_{mn}}$ and take the diagonal entries in each block as:

$$\triangle \equiv \delta_{block}((I \otimes F)T_{mn}(I \otimes F)^*), \tag{3.56}$$

where the operator $\delta_{block}(\cdot)$ denotes taking the diagonal entries in each block and can be expressed as:

$$\delta_{block}(\cdot) \triangleq (I \otimes I)(\cdot)(I \otimes I),$$

and then the best circulant approximation $c_F(T_{mn})$ for the BTTB matrix T_{mn} is obtained via using two FFTs to $\triangle$ block wisely:

$$c_F(T_{mn}) = (I \otimes F)^* \triangle (I \otimes F). \tag{3.57}$$

It is worth to noting that in the PCG routine, we need to solve the linear system $c_F(T_{mn})\tilde{x} = \tilde{y}$ for $\tilde{x}$ given $c_F(T_{mn})$ and $\tilde{y}$ in each iteration. This requires the preconditioning matrix $c_F(T_{mn})$ to be well structured. Let $(T_{mn})_{i,j;k,l} = (T_{k,l})_{i,j}$ be the (i,j)-th entry of the (k,l)-th block in BTTB matrix T_{mn}, then define P be the permutation matrix that satisfies

$$(P^* T_{mn} P)_{k,l;i,j} = (T_{mn})_{i,j;k,l},$$

Algorithm 8 Multi-Splitting Methods for TV-ℓ_1 with Preconditioning

Require: $A, b, u^0, y^0, \alpha, \beta, r > 0, \rho > 0, tol > 0$

Set $\mathbf{T}$ via (3.49), $\mathbf{f}$ via (3.50)
Set $k \leftarrow 0$,
while "not converge" **do**
  Compute w^{k+1} according to (3.16).
  Compute v^{k+1} according to (3.17).
  while "not converge" **do**
    Compute u^{k+1} according to (3.55) via PCG
    with preconditioner $c_F(T_{mn})$ given in (3.57).
  end while
  Compute y^{k+1} according to (3.19).
  if $\tau \leq tol$ **then**
    Return u^{k+1}
  else
    Set $k \leftarrow k+1$
  end if
end while

where $1 \leq i, j \leq n$, $1 \leq k, l \leq m$. Note that

$$P^* \triangle P = \begin{pmatrix} \widetilde{T_{1,1}} & 0 & \cdots & 0 \\ 0 & \widetilde{T_{2,2}} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \widetilde{T_{m,m}} \end{pmatrix}, \tag{3.58}$$

where $\widetilde{(T_{k,k})}_{ij} = (\delta(FT_{i,j}F^*))_{kk} = (\delta(FT_{(i-j)}F^k))_{kk}$, for $1 \leq i, j \leq n$, $1 \leq k \leq m$. Thus $(c_F(T_{mn}))^{-1} = [(I \otimes F^*)P](P^*\Delta P)^{-1}[P^*(I \otimes F)]$ and the solution $\tilde{x}$ can be expressed as

$$\tilde{x} = (c_F(T_{mn}))^{-1}\tilde{y}. \tag{3.59}$$

Now we can add the PCG feature to the proposed algorithm and summarize it as Algorithm 8. However, in practice, solving the linear system (3.59) or finding $(c_F(T_{mn}))^{-1}$ is also expensive, since $c_F(T_{mn}$ is a block circulant with circulant block (BCCB) matrix and still expensive for computing its inverse. However, in this case, we can still take its advantage in the circulant block structure to implement a fast matrix vector product to accelerate the computation, instead of solving the (3.59) system or finding its inverse directly.

Chapter 4

Numerical Results and Application in MRI

In this chapter, we demonstrate the effectiveness of **Algorithm 8** for the case of image reconstruction with partial Fourier data as well as its performance in the application of sparse MR imaging comparing it with the existing packages SparseMRI V.02 [MJ07a] and RecPF V.1.1 [YY08]. We mainly solve the model (3.10) where the underdetermined system $Ax = b$ is the constraint condition that forces the underlying variable to fit the fidelity requirement. b is generated from an image u which possesses transformed sparsity in wavelet domain. In particular, the sensing matrix $A \in \mathbb{R}^{M\times N}$, where $M < N$, K is the sparsity level of the transformed image, and

$$b = A(u + n_1) + n_2, \tag{4.1}$$

where n_1 and n_2 are the additive Gaussian noise vectors whose component's are $i.i.d$ distributed as $N(0,\sigma_1^2)$ and $N(0,\sigma_2^2)$. Both the original image and the measurement b can be corrupted by noise.

The sensing matrix A in our case is a partial Fourier matrix, composed by a selection operator and Fourier matrices, that is $A = PF$, where F is an $N \times N$ Fourier matrix and the M rows in the selection operator P are chosen randomly from N rows of an $N \times N$ identity matrix $\mathbf{I}$ or from the rows of $\mathbf{I}$ indicated by the indices generated along a particular sampling trajectory. Both two types of matrices are good matrices for Compressive Sensing. While the partial Fourier matrix A is stored implicitly, fast matrix vector multiplication is applicable via the fast Fourier transform (FFT) with the cost of $O(N\log N)$ flops. Furthermore, the partial Fourier matrix A has orthonormal rows such that $A^TA = \mathbf{I}$.

This chapter is organized as follows. We first test the numerical performance of our proposed algorithm with a series of numerical experiments, which include a

test of robustness to the choice of the regularization parameter r, show the role of the parameters r and ρ in determining the convergence rate, demonstrate the BTTB preconditioning and the recoverability of the proposed algorithm. In the second part, we mainly focus on the application of the proposed algorithm in MR imaging and show the results of a comparison with existing packages in reconstructing the MR images by using real clinical MR data.

4.1 A test on the proposed algorithm

The performance of compressive sensing algorithms varies with the change of the parameters, such as, the number of measurements M, the size of the image N and the sparsity level K of the image under the transformed domain. In this section, we test the numerical performance of **Algorithm 8** and record the results under various parameter combinations of interest. The numerical results confirm the theoretical analysis in the performance of BTTB preconditioning and the convergence rate with respect to various ρ values; besides, the robustness of the results under various r values can also be determined from to its numerical performance in numerical experiments.

We set the initial iterate to $u^0 = A^T b$, since this value contains the problem specific information and is easy to calculate. Further, u^0 is also a feasible point that minimizes the Euclidean norm of the equality constraint $\frac{1}{2}||Au-b||^2$ when A is orthonormal. Although all nonnegative values of the Lagrangian multiplier $y \in \mathbb{R}^M$ works for the problem, we initialize $y^0 = [1, \cdots, 1]^T$ for simplicity. In all the experiments, the ratio of weights of ℓ_1 and TV regularization in (3.10) is denoted as $\eta = \frac{\alpha}{\beta}$, the sampling rate is denotes as $\delta = \frac{M}{N}$, these parameter combinations will be tested in our experiments.

Our code is written in MATLAB (Release 7.11.0), and all experiments were performed on a Lenovo Think Pad T61p workstation with Intel dual core T7300 2.0GHZ CPU and 2GB RAM. The objective of our interest in this section is restoring the Shepp-Logan phantom. The partial Fourier data are collected in the frequency domain along

Figure 4.1: From left: (1) The sampling pattern of 22 radial lines in K-space; (2) The 256 by 256 Shepp-Logan phantom.

radial lines Figure 4.1. Other sampling patterns and more complicated images will be considered in the numerical experiments in next section.

4.1.1 The choice of ρ and rate of convergence

The choice of ρ in **Algorithm 8** depends on the eigenvalue distribution of the matrix operator $\mathscr{H}(A^TA)^{-1}$, where $\mathscr{H} \triangleq I + D^TD + A^TA$. When $\rho_{opt} = 2r\frac{\lambda_m\lambda_M}{\lambda_M+\lambda_m}$ and $\frac{\lambda_m\lambda_M}{\lambda_M+\lambda_m} \leqslant 1$, the optimal linear convergence rate is reached at ρ_{opt}. Particularly, in Compressive Sensing applications the sensing matrix A is orthonormal, that is $A^TA = I$. Then the relative convergence rate depends on the spectrum of the matrix $2I + D^TD$. The necessary condition of convergence $0 < \rho \leqslant 2r$ implied by Theorem 3.2.2 can be relaxed to $0 < \rho \leqslant 2r(2+\gamma_i)$, where $\gamma_i \triangleq eig(D^TD)_i$, $i = 1, \cdots, N$. Hence ρ is bounded by $\rho_{max} = 2r(2+\gamma_{max})$, and (3.38) is equivalent to:

$$\bar{Y}^{k+1} = (I - \frac{\rho}{r}(2I + D^TD)^{-1})\bar{Y}^k, \tag{4.2}$$

Since the matrix $2I + D^TD$ is symmetric and its eigenvalue decomposition can be written as:

$$2I + D^TD = Qdiag\{2+\gamma_1, \cdots 2+\gamma_N\}Q^T,$$

Table 4.1: The eigenvalue of $D^T D$

	16×16	100×100	400×400	1600×1600
γ_{max}	8	8	8	8
γ_{min}	0	0	0	0

where Q is an orthonormal matrix. By multiplying A^T at both side of (4.2) and rearranging the orthonormal matrices A and Q we have that:

$$QA^T\bar{Y}^{k+1} = \Lambda_\gamma(\rho)QA^T\bar{Y}^k,$$

where $\Lambda_\gamma(\rho) \triangleq I - \frac{\rho}{r}diag\{\frac{1}{2+\gamma_1},\cdots\frac{1}{2+\gamma_N}\}$. Then define $\tilde{Y}^{k+1} \triangleq QA^T\tilde{Y}^k$, then we have

$$|\tilde{Y}_i^{k+1}| \leqslant |\Lambda_{\gamma_i}(\rho)||\tilde{Y}_i^k|,$$

where the growth factor $\Lambda_{\gamma_i}(\rho) \triangleq |1 - \frac{\rho}{r(2+\gamma_i)}|$. Then the optimal choice of ρ can be determined in a similar way as we showed in (3.40), that is

$$\rho_{opt} = 2r\frac{(2+\gamma_{min})(2+\gamma_{max})}{4+\gamma_{min}+\gamma_{max}}. \tag{4.3}$$

According to the numerical results in Table (4.1), $\gamma_{max} = 8$ and $\gamma_{min} = 0$ for any matrices $D^T D$ with even number of columns and rows, that is $\rho_{opt} = 2r\frac{2(2+8)}{4+8} \backsimeq 3.3r$ and $\rho \in (0, 20r]$. So far we present a perfect theoretical analysis on how the value of ρ affects the convergence rate, but how does it perform in practical computation? In the following, we test the algorithm 8 and present the numerical results to support the theoretical results.

To test the role of ρ on how it affects the rate of convergence, we solve the image reconstruction problems using 19% and 27% partial Fourier data in various dimensions for a wide range of $\rho \in (0, 20r]$ with respect to the fixed $r = 1e2$. We set the stopping tolerance $\tau = 10^{-2}$ and fix the regularization parameter $r = 1e2$. The experiments are conducted for the combination of four sizes of problems and the various ρ values from

Table 4.2: The rate of convergence w.r.t various ρ values against each dimensionality

	$r = 1e2, \quad \delta = 0.19, \quad \eta = 0.5, \quad \tau = 10^{-2}$											
	$\rho = 0.5r$			$\rho = r$			$\rho = 1.5r$			$\rho = 2r$		
N	Iter	Err	Time	Iter	Err	Time	Iter	Err	Time	Iter	Err	Time
64^2	57	0.162	5.32	55	0.160	5.18	54	0.160	5.12	54	0.159	4.70
128^2	41	0.040	8.98	38	0.038	8.16	37	0.038	8.25	37	0.036	7.92
256^2	35	0.023	29.34	30	0.020	25.58	28	0.020	24.44	27	0.019	22.59
512^2	32	0.016	92.18	27	0.015	83.63	25	0.014	77.51	23	0.013	71.36
	$\rho = 3r$			$\rho = 6r$			$\rho = 15r$			$\rho = 20r$		
64^2	53	0.162	5.27	53	0.160	5.17	52	0.164	5.44	–	–	–
128^2	36	0.038	8.26	35	0.040	7.76	35	0.041	8.29	–	–	–
256^2	26	0.018	20.58	25	0.019	20.35	24	0.021	20.02	–	–	–
512^2	22	0.012	68.64	20	0.012	64.28	19	0.014	61.94	–	–	–
	$r = 1e2, \quad \delta = 0.27, \quad \eta = 0.5, \quad \tau = 10^{-2}$											
	$\rho = 0.5r$			$\rho = r$			$\rho = 1.5r$			$\rho = 2r$		
N	Iter	Err	Time	Iter	Err	Time	Iter	Err	Time	Iter	Err	Time
64^2	45	0.049	4.23	44	0.044	4.03	43	0.045	3.36	42	0.047	3.87
128^2	29	0.033	6.10	27	0.022	5.67	26	0.019	5.30	26	0.018	4.65
256^2	26	0.021	19.7	22	0.020	16.20	21	0.017	15.26	20	0.014	15.72
512^2	25	0.015	72.04	21	0.016	62.08	20	0.014	61.39	19	0.012	57.71
	$\rho = 3r$			$\rho = 6r$			$\rho = 15r$			$\rho = 20r$		
64^2	44	0.050	4.47	43	0.054	4.26	43	0.054	4.38	–	–	–
128^2	25	0.018	5.44	24	0.020	5.55	24	0.002	5.25	–	–	–
256^2	19	0.011	14.47	18	0.012	14.26	43	0.013	30.79	–	–	–
512^2	19	0.008	59.5	15	0.008	50.28	37	0.009	108.37	–	–	–

$0.5r$ to $20r$ and we record the number of iterations to converge, the relative error and the CPU time.

The numerical results in Table 4.2 show that when $\rho = 20r$ the algorithm 8 diverges since. But on the other hand, we ρ is too small, such as when $\rho = .5r$, the convergence rate becomes slowe and keep increasing as ρ approaches to $15r$. In this work we suggest the optimal $\rho \in [3r, 6r]$, since as we may read from the table the convergence rate does not keep increasing as we assign larger ρ value to it. For example, in the 512^2 reconstruction case, when ρ is greater than $6r$, the processing time get increased dramatically. Actually according to our theoretical analysis results $\rho_{opt} = 3.3r$, and this result matches our numerical resluts shown in the table very well.

4.1.2 The sensitivity to r

Intuitively, we know that r should be proportional to $\frac{1}{2}||Au-b||^2$ and the method of continuation [YY08] for updating r proves to be an efficient way of updating the regularization parameter. On the one hand, r is used as a penalty of the fidelity and a large r is preferred to reduce the noise. On the other hand, a large r may lead to a longer computation to reach the desired stopping criteria. In this part, we study the sensitivity of r to the accuracy of the solution. Furthermore, we show how algorithm 8 performs for all r values of interest via a series of numerical experiments.

Table 4.3: The sensitivity to r values

	$N=64^2,\ \delta=0.31,\ \eta=0.5,\ \tau=1e{-}3,\ SNR=4.5dB,\ \rho=2r$										
r	2^3	2^4	2^5	2^6	2^7	$1e2$	$2e2$	$5e2$	$8e^2$	$1e^3$	$2e3$
Iter	49	55	51	52	61	62	87	200	312	386	598
Err	0.306	0.163	0.088	0.046	0.021	0.029	0.014	0.007	0.007	0.007	0.008
Obj	924.4	816.4	760.7	730.5	719.1	719.8	713.7	707.0	705.4	705.6	704.7
Time	4.27	4.88	4.58	4.63	5.56	5.39	8.17	18.15	28.48	35.69	55.03
Fid	0.031	0.028	0.028	0.032	0.052	0.029	0.042	0.031	0.028	0.027	0.027
	$N=128^2,\ \delta=0.31,\ \eta=0.5,\ \tau=1e{-}3,\ SNR=7dB,\ \rho=2r$										
r	2^3	2^4	2^5	2^6	2^7	$1e2$	$2e2$	$5e2$	$8e^2$	$1e^3$	$2e3$
Iter	39	37	37	36	39	35	48	86	132	162	309
Err	0.150	0.081	0.043	0.022	0.012	0.015	0.008	0.004	0.004	0.004	0.005
Obj	2632.6	2323.9	2157.8	2067.2	2018.8	2032.6	2003.4	1992.8	1989.8	1989.2	1990.7
Time	7.75	7.80	7.78	7.58	8.26	6.93	10.00	17.90	26.91	33.42	64.73
Fid	0.059	0.058	0.059	0.063	0.064	0.070	0.06	0.06	0.05	0.05	0.05

The value of r controls the fitness of the data and balances the tradeoff between the fidelity and sparsity level. In [SD07], the author suggests a lower bound on r, $r \geqslant \frac{1}{||A^T b||_\infty}$, since when $r < \frac{1}{||A^T b||_\infty}$, $\mathbf{0}$ becomes an unique optimal value of the ℓ_1 least squares problem. In the context of our algorithm, r controls the shrinkage level in (3.16) and (3.17), that is all the components less than $\frac{1}{r}$ are shrunk to zero. A small r value may make the estimation lose fidelity, but an overly large r value may lead to a huge computation and increased CPU time, since only a few components are shrunk to zero in each iteration and many more steps are required such that the solution reaches a desired sparsity level.

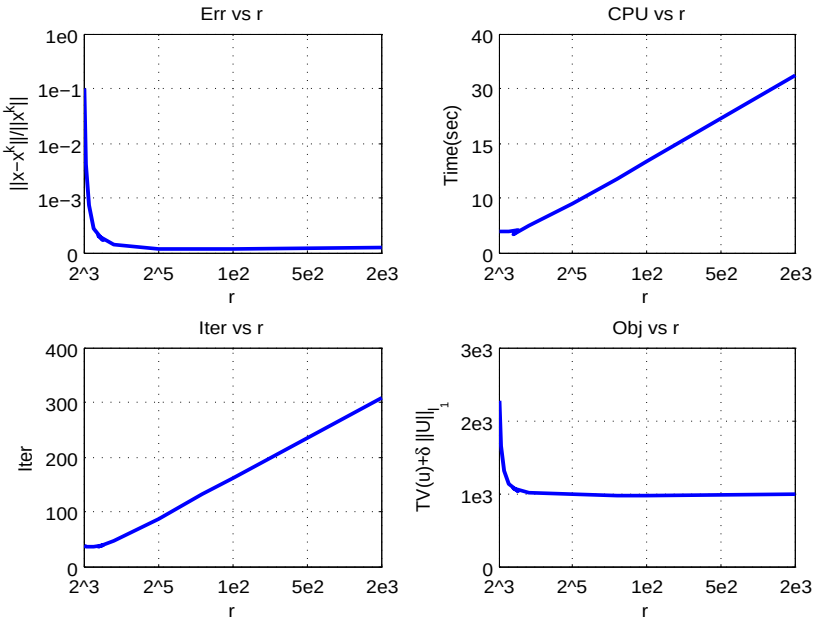

Figure 4.2: The relative error $\frac{||x-x^*||}{||x^*||}$, number of iterations to converge, CPU time and value of objective function versus various r for reconstructing 128^2 image using 31% partial data via Algorithm 8.

To test the robustness of r, we solved the partial reconstruction problem for a wide range of values of r. We set $\rho = 2r$ and use $\delta = 31\%$ partial Fourier data to reconstruct the 64^2 and 128^2 phantom Shepp-Logan image by varying the r value from 2^3 up to $2e3$. The experiments compare the performance of algorithm 8 with respect to various regularization parameters. All data points of the numerical experiments listed in Table 4.3 represent an average over 5 runs.

A representative sample of experimental results are shown in Figure 4.4, which depicts the relative error, CPU time, number of iterations to convergence and and the value of the objective function for reconstructing the 128×128 Shepp-Logan phantom image using 31% partial Fourier data under all r values of interest.

Several conclusions may be drawn from this figure. First, the Algorithm 8 is

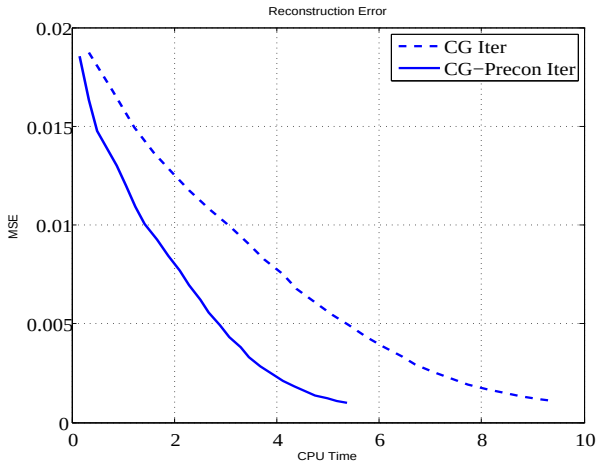

Figure 4.3: against the CPU time for reconstructing 128×128 phantom by using 25% partial Fourier data

robust to r and produces accurate reconstructions with respect to a wide range of r. But the CPU time and number of iterations are increasing almost linearly as with r, such as when r changes from 2^7 to $2e2$, the CPU time increased by 20%.

Second, we are interested in finding an optimal r value from the experimental data. A desired r value should be able to bring enough accuracy within an economic computational effort. According to Table 4.3 and Figure 4.4, we find that $r = 1e2$ is an ideal one since comparing with other columns in the table, $r = 1e2$ provides a high accuracy with an acceptable computational work.

4.1.3 The BTTB preconditioning and its implementation

As we know, we can accelerate the conjugate gradient method by improving the eigenvalue distribution of the iteration matrix via applying a preconditioner. Instead of solving the linear system $Hx = b$ directly, we need to solve the rescaled system $\hat{H}\hat{x} = \hat{b}$, where $\hat{H} = C^{-T}HC^{-1}$, $b = C^{-1}b$, $\hat{x} = Cx$ and C is the so called preconditioner. A well designed preconditioner C should be able to make the spectrum of $C^{-T}HC^{-1}$ clustered, and possess a simple structure such that $Cx = \hat{x}$ is easy to solve.

Actually, the preconditioning process PCG used in Algorithm 8 require to solve a linear system for $\tilde{x}$ depicted in (3.59) in each iteration. However, in practical implementation, we can take advantage of the block circulant structure in the circulant block preconditioner $c_F(T_{mn})$, and apply a fast matrix vector multiplication to accelerate the computation, instead of using the inverse of the preconditioner $c_F(T_{mn})$ shown in (3.59). Specifically, the j-th $n \times n$ block in the first block column of BCCB $c_F(T_{mn})$ is a circulant matrix, denoted as C_j, where $j = 0, \cdots, m-1$. C_j can be diagonalized by two FFTs, $\Lambda = FC_jF^*$, where F is the $n \times n$ Fourier matrix, $\lambda_k = \Lambda_{kk}$ is an eigenvalue of C_j. The product of C_jv and $C_j^{-1}v$ can be computed easily via FFTs as $C_jv = F\Lambda F^*v$ and $C_j^{-1}v = F\Lambda^{-1}F^*v$, respectively, within $O(n\log n)$ operations. The product of $c_F(T_{mn})v$ can also be processed in a similar way with respect to each block. Here we need to point out that there are at most m different blocks in the BCCB matrix $c_F(T_{mn})$ which has $m \times m$ blocks. Thus $c_F(T_{mn})v$ can be computed with $O(mn\log mn)$ operations. Figure 4.3 shows a comparison of the CPU time between a preconditioned system and a system without preconditioning.

4.2 The MR imaging application

Compressive Sensing (CS) aims to reconstruct a signal or image by using fewer measurements than required in traditional way while not degrading the quality. This feature is attractive and the CS application has rapidly spread to many areas which are related with restoring the sparsity. Magnetic Resonance Imaging (MRI) becomes one of the most important tools used in modern clinical diagnosis, and the imaging speed is an obstacle is MRI due to some fundamental limits in physical and physiological constraints [MJ07b]. CS becomes a potential way to reduce the scanning time without reducing the quality of the image by using less data.

In this section, we mainly discuss the application of Compressive Sensing in MR image reconstruction. Some early works on the CS application in MRI are found

in [MJ07a][MJ07b][SD07][WY08]. In the following, we first state the principles of MR imaging and CS applications in MRI and this part is mainly based on M.Lustig's work [MJ07b]; in the second part, we show the application of our algorithm in reconstructing MR images and compare the numerical performance with other packages: RecPF [YY08], and SparseMRI [MJ07a].

4.2.1 MR imaging principles and CS in sparse MRI

The MRI signal is generated by the frequency response of tissues in the body, mostly those in water molecules. First a strong static magnetic field is applied and the protons are polarized while yielding a net magnetic moment oriented in the direction of the static field. Then a radio frequency (RF) pulse is applied and a magnetization component is produced transverse to the static field. At the same time, the protons in the area where the RF is applied get aligned along the magnetic field direction and spin with a certain frequency. When the RF pulse is turned off, the protons return to their natural state and release a signal. Here the transverse magnetization at position l is represented by the complex quantity

$$m(l) = |m(l)|e^{-i\phi(l)},$$

where $|m(l)|$ is its magnitude and $\phi(l)$ represents the phase which indicates the direction of the magnetization pointing in the transverse plane. The MR image we are interested in is $m(l)$ depicting the spatial distribution of the transverse magnetization. Actually the signal received by the external coil when the RF pulse is turned off can be obtained as the integration over the entire volume:

$$s(t) = \int_R m(l)e^{-i2\pi k(t)l}dl,$$

where the received signal $s(t)$ is the Fourier transform of the object $m(l)$ sampled at the spatial frequency $k(t)$. In other words, the MR image of spatial energy in reconstructed from data acquired in the frequency domain or the so called K-space, and this is different from traditional optical imaging where pixel samples are measured directly.

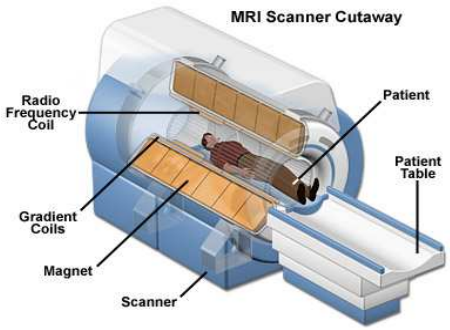

Figure 4.4: The MRI machine

Constructing a single MR image commonly involves collecting a series of frames of data along a trajectory in k-space, called data acquisitions. The image resolution is mainly determined by the size of the sampled region in k-space. Generally a larger sampling region gives higher resolution; the supported field of view (FOV) is determined by the sampling density in k-space, and generally a larger objects require a denser sampling to meet the Nyquist rate. Violation of Nyquist rate will cause artifacts in the reconstruction. The sampling pattern or the k-space trajectory for data acquisition is also a source affecting the reconstruction quality. So far the most popular trajectory used in clinical imaging the straight lines from a Cartesian grid. The reconstruction for this sampling pattern is very simple and can be achieved via inverse Fast Fourier Transform (IFFT). Besides other sampling patterns are also used, including the sampling along the radial lines (Figure 4.1) and along the spiral trajectories. Radial acquisition are less susceptible to motion artifacts than Cartesian trajectories [GJ92] and can be significantly under-sampled [KJ98]. So this fits the CS features very well and our partial reconstruction based on it. Spirals make efficient use of the gradient system hardware, but such non Cartesian trajectories are more complicated, requiring a k-space interpolation scheme, e.g. griding [JA91].

The data acquisition process becomes the main obstacle to imaging speed. And since the sampling speed is limited by physical constraints, reducing the amount of

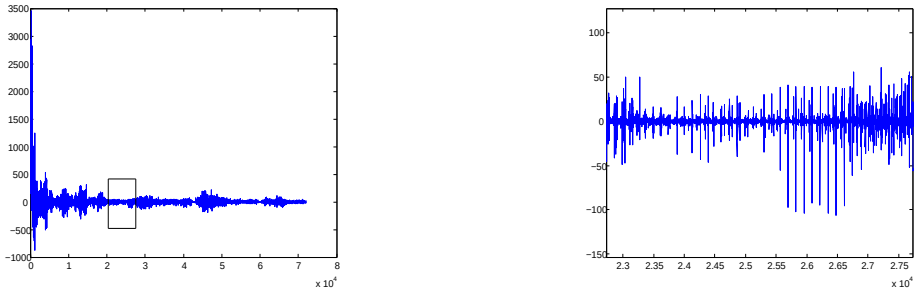

Figure 4.5: The wavelet coefficients for 256×256 Cameraman image (left), the sparsity zoom in a window (right)

required data becomes a possible way to reduce imaging speed. Therefore compressive sensing application in MRI is a potential way to reduce the amount of data without hurting the quality of the image. Basically, a successful CS application needs to meet several requirements, which include sparsity or transformed sparsity structure in the underlying objective, the incoherence sampling pattern to the sparsity transformation, and an efficient nonlinear reconstruction algorithm which should be able to enforce the sparsity and reconstruct the image in an economic way.

Actually, the MR imaging fits the CS requirements very well. First, the sparsity of most of MR images are successfully realized by representing the image in an appropriate transform domain, such as wavelets (Figure 4.5). As we know, a natural image can be mapped into a vector of sparse coefficients. The image can be approximated by the linear combination of the most significant coefficients while ignoring the smaller ones. Second, although the coherence is very low for the full random sampling, sampling a truly random subset of k-space is generally impractical for the hardware and physiological constrains. On the other hand, most of the energy in MR images are concentrated around the center of k-space and rapidly decays towards the periphery (Figure 4.7), and uniform random sampling does not take this into account. So realistic sampling should be denser around the center in k-space, and the radial lines trajectory matches this very well. For more detailed discussion on incoherence we re-

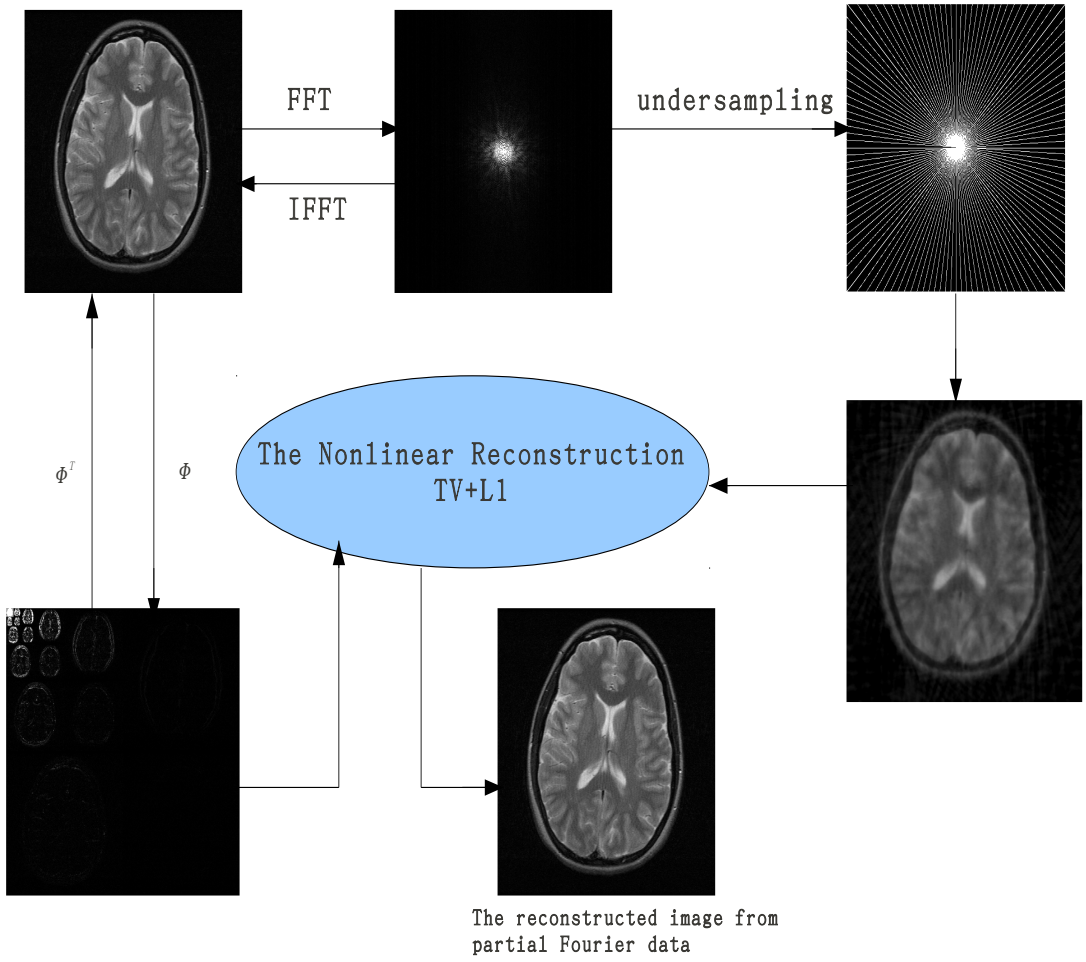

Figure 4.6: Instead of taking the full set of sample in K-space, the partial sampling is taken along the trajectory which has more density in center and less density at outside, and use its back-projection as the observation and inputed into the nonlinear solver, and by minimizing the wavelet coefficients and TV of the impinge we finally have an image reconstructed via partial data without hurting the quality.

fer [MJ07a]. Third, the main theme of this work is developing an efficient numerical algorithm for preserving the sparsity of the underlying objective. In the following, we will demonstrate the numerical performance of the proposed algorithm in reconstructing MR images. A fully numerical comparison with other existing packages is also presented.

4.2.2 Comparison with RecPF and SparseMRI packages in reconstructing MR images using partial Fourier data

For the convenience of our discussion, we give Algorithm 8 the name ALSR (Augmented Lagrange Sparse Reconstruction). In this section, we present numerical simulations on reconstructing the MR images with the ALSR as well as on the numerical performance of two other existing algorithms: a nonlinear conjugate gradient method based algorithm SparseMRI [MJ07a], and alternating direction method based algorithm RecPF [YY08]. Both methods are regarded as efficient Compressive Sensing algorithms for reconstructing MR images from partial Fourier data, specifically SparseMRI is an early application of CS in MR imaging and RecPF is a fast algorithm possessing the speed of FFTs. All three algorithms ALSR, SparseMRI and RecPF can be used to solve the TV-ℓ_1 regularized model:

$$\begin{aligned} \min_{u} \quad & \alpha||\Psi^T u||_{\ell_1} + \beta TV(u) \\ s.t. \quad & F_p u = b, \end{aligned} \tag{4.4}$$

where F_u denotes an orthonormal partial Fourier operator, that is $F_u^T F_u = I$.

SparseMRI aims to solve the equivalent unconstrained version of (4.4) with regularization parameter r as:

$$f(u) \triangleq \alpha||\Psi^T u||_{\ell_1} + \beta TV(u) + \frac{r}{2}||Au-b||^2, \tag{4.5}$$

with a nonlinear conjugate gradient method with backtracking line search, where the

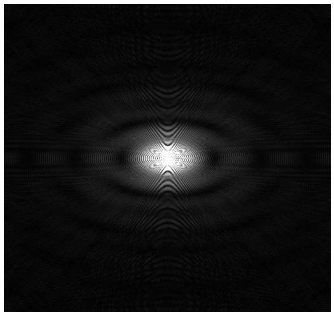

Figure 4.7: The energy in MR images are concentrated around the center of k-space and rapidly decays towards the periphery

gradient of (4.5) is expressed as:

$$\bigtriangledown f(u) = \alpha \bigtriangledown ||\Psi^T u||_{\ell_1} + \beta \bigtriangledown TV(u) + rA^T(Au-b),$$

In practical implementation the absolute value is approximated wit a smoothing parameter μ, for instance $|x| \approx \sqrt{x^2+\mu}$ and its gradient can be expressed as $\frac{d|x|}{x} \approx \frac{x}{\sqrt{x^2+\mu}}$, where $\mu \in [10^{-15}, 10^{-6}]$. SparseMRI relaxes the non-differentiability of the TV and ℓ_1 via a smoothing parameter and terminates when the Euclidean norm of the gradient $|| \bigtriangledown f|| \leqslant tol$. The implementation enjoys a simple process, but this algorithm performs much slower than the other two methods. RecPF is a fast algorithm for solving the TV-ℓ_1 regularized problem (4.4) based on the alternating direction method. Start from the UN-constraint problem (4.5), by introducing slack variables while penalizing the discrepancy between the slack variables and the TV and ℓ_1 terms respectively, the unconstrained objective function is separated into several subproblems with closed form solution that can be represented via 2D shrinkage. After each slack variable is updated, u can be updated with respect to fixed values of slack variables. The routine will keep updating the slack variables and the objective variable u alternatively till the stopping criteria is satisfied. Mathematically, denote the 2-D shrinkage as $s_r(y) = \frac{y}{||y||}.*max\{|y|-r,0\}$. The slack variables v,w can be updated via $v^{k+1} = s_r(D^T u^k)$ and $w^{k+1} = s_r(|\Psi^T u^k|)$ with fixed value of u^k. Then one solves for u with the updated slack

variables w^{k+1} and v^{k+1} from a quadratic problem $u^{k+1} = arg\min_u Q_r(u\ ; v^{k+1}, w^{k+1})$ that minimizes the total discrepancy.

We need to point out that both RecPF and ALSR are operator splitting methods in updating the variables alternatively and take advantage of the fast shrinkage operator in computation. RecPF processes the image without breaking its structure so that it can take advantage of fast 2D operations in Matlab implementation, especially u^{k+1} can be solved via three FFTs by taking the advantage of structured circulant iteration matrix. But if the image is vectorized and processed as a one dimensional signal, RecPF encounters difficulties and the circulant structure will not exist in this case. ALSR is generally based on the augmented Lagrangian methods. The step for updating the dual variable leads to a way for analyzing the convergence rate, and an optimal rate of convergence can be obtained theoretically. Besides ALSR processes the image in a more general way of vectorizing it as a one dimensional signal and each pixel can be processed in parallel. In the step for solving for u, the iteration matrix is a BTTB matrix and an optimal block wise circulant approximation is used as a preconditioner in the conjugate gradient routine and a fast matrix vector multiplication is achieved via a FFTs. Last, the choice of stopping criteria used in RecPF requires to evaluate the sub-differential of each subproblem to terminate the routine when their maximum value falls to a certain level, while ALSR simply terminates the routine if the function can not bring any striking decrease.

Experiment 1: Brain reconstruction

Brain scans are the most common clinical application of MRI, which can detect a variety of conditions of the brain such as cysts, tumors, bleeding, swelling, developmental and structural abnormalities, infections, inflammatory conditions, or problems with the blood vessels. It can determine if a shunt is working and detect damage to the brain caused by an injury or a stroke. MRI of the brain can also be useful in evaluating prob-

lems such as persistent headaches, dizziness, weakness, and blurry vision or seizures, and can help to detect certain chronic diseases of the nervous system, such as multiple sclerosis. In some cases, MRI can provide clear images of parts of the brain that can not be seen as well with an X-ray, CAT scan, or ultrasound, making it particularly valuable for diagnosing problems with the pituitary gland and brain stem.

In this part, our main objective is to test the application of CS to brain images collected in a clinic. Figure 4.8 (a) and Figure 4.9 (a) are the 256×256 and 512×512 brain images of a full Nyquist sampled data set. We compare the performance of three packages sparseMRI, RecPF and ALSR on reconstructing the two brain images with partial Fourier data collected in the k-space along the radial lines depicted in Figure 4.8 (b) and Figure 4.9 (b). As we know, CS can be used to reduce the amount of the sampling without hurting the quality of the images when the requirements of the CS are satisfied. For the applications here, the brain images shows a transformed sparsity in wavelet domain as Figure 4.5. On the other hand, according to the result in Figure 4.8 (d)-(f) and Figure 4.9 (d)-(f), the sampling pattern of radial lines matches the characteristics of k-space frequency distribution very well although the incoherence is hard to prove.

From Figure 4.8 and Figure 4.9, the three packages works well on partial reconstruction, the last rows in both two figures focus on a special part of the brain in the reconstructed images, and from the images in the second row from the last, we can easily identify the improvement on the contrast of the tissues, the denoising as well as the deblurring. These two figures are the reconstruction with 28% partial Fourier data, although the quality of the reconstruction in each package is similar, the processing times are significantly different. SparseMRI requires 48.3s for restoring a 256×256 images and 152.3s for restoring a 512×512 image, but ALSR and RecPF shows a much better performance with respect to speed. These two require only around 6s for restoring a 256×256 image and 24s for the 512×512 images. This is mainly because

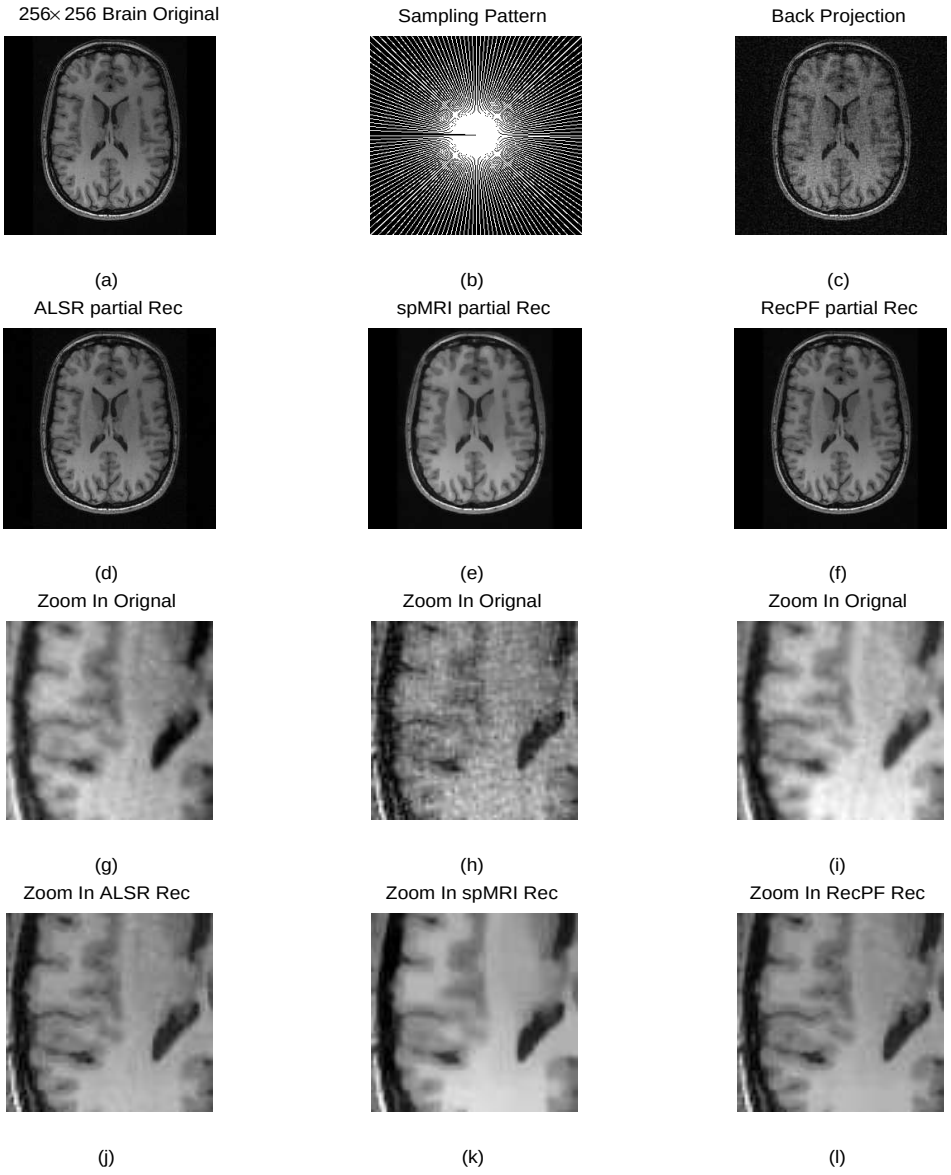

Figure 4.8: Figure (a)-(l) depicts a 256×256 Brain Imaging Reconstruction via the packages ALSR, sparseMRI and RecPF. We take 28% partial Fourier data in the k-space along radial lines as (b), the back projection image (c) has $SNR = 9dB$, and the SNR in reconstructed images (d)-(f) are enhanced to $SNR = 32dB$ within 6.1s, 48.3s and 6.2s via ALSR, sparseMRI and RecPF respectively. Zoom in part of the back projection of the brain with additional Gaussian background noise, we see that all three packages reconstruct the image from partial data with comparable visual quality, the contrast are enhanced the noise level is reduced to $SNR = 32dB$.

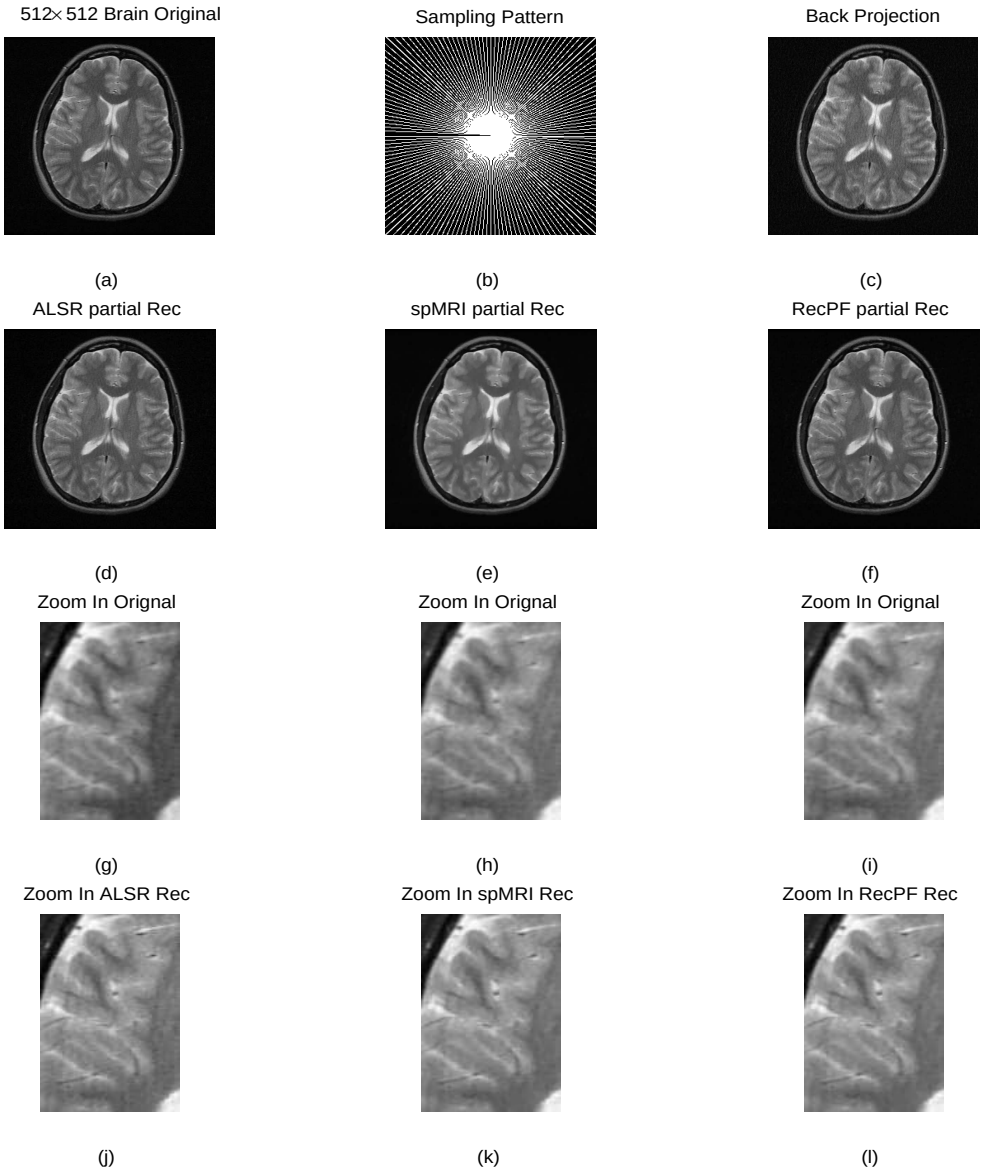

Figure 4.9: Figure (a)-(l) depicts a 512 $\times$ 512 Brain Imaging Reconstruction via the packages ALSR, sparseMRI and RecPF. We take 28% partial Fourier data in the k-space along radial lines as (b), the back projection image (c) has $SNR = 9dB$, and the SNR in reconstructed images (d)-(f) are enhanced to $SNR = 28dB$ within 24.2s, 152.3s and 23.1s via ALSR, sparseMRI and RecPF respectively. Zoom in part of the back projection of the brain with additional Gaussian background noise, we see that all three packages reconstruct the image from partial data with comparable visual quality, the contrast is enhanced, and the noise level is reduced, yielding $SNR = 28dB$.

the nonlinear conjugate gradient method converges much slower, and the high nonlinearity of the TV operator also slows down the gradient methods, while the operator splitting methods show favorable properties for these large scale problems.

Experiment 2: 3D Angiography Reconstruction

Magnetic resonance angiography (MRA) is a technique based on Magnetic Resonance Imaging (MRI) mainly used to image blood vessels, such as the images of the arteries in order for evaluating them for stenosis, occlusion or aneurysms. MRA is often used to evaluate the arteries of the neck and brain, the thoracic and abdominal aorta, the renal arteries, those in arms and legs. Traditionally, to enhance the contrast of the vessels and blood, before the MR scanning the patient needs to be injected a MRI contrast agent and images are acquired during the first pass of the agent through the arteries.

The CS is particularly suitable for angiography. Since in angiography there are only bright areas in blood vessels and a very low background signal, it appears to be sparse in the image domain. Furthermore it also shows a well transformed sparsity under the wavelet transformation and finite differences. On the other hand, since the angiography often needs to cover a very large FOV with relative high resolution, this will be a time consuming process and the amount of the collected data for reconstructing the images of this kind is huge. Hence MRA requires a scheme of under-sampling to save scanning time and a fast sparsity enforcing algorithm to enhance the contrast and preserve the sparsity within the acceptable time.

In this part, we mainly test the behavior of the proposed algorithm ALSR and the CS applications in reconstructing the angiograms with respect to various under-sampling rates. The numerical experiment aims to reconstruct an angiograms of the peripheral legs via under-sampling a full Nyquist rate MR data set collected in k-space, the data matrices was set to $128 \times 128 \times 64$ with corresponding resolution of $1 \times 0.8 \times 1\ mm$.

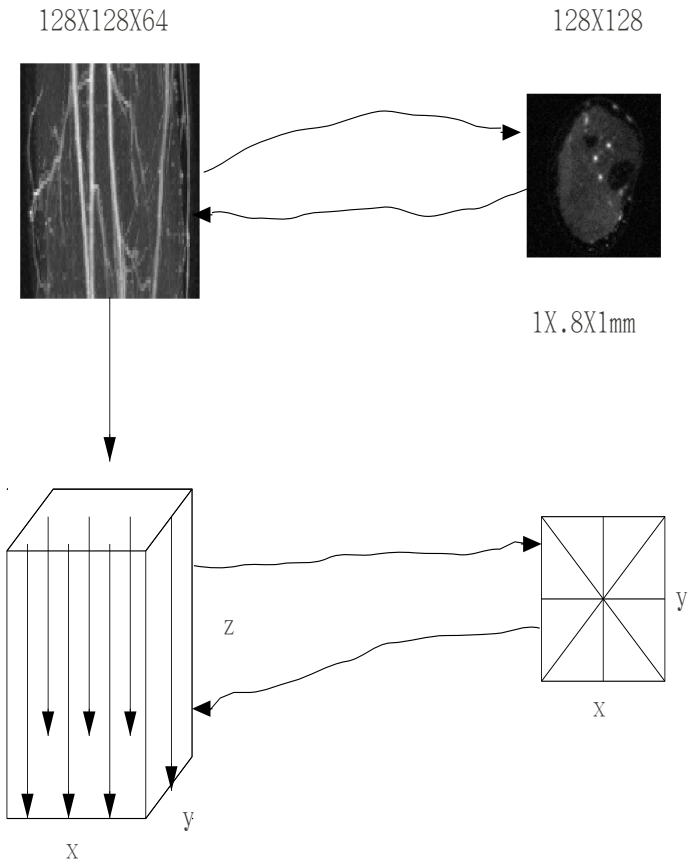

Figure 4.10: The $128 \times 128 \times 64$ 3D Angiography reconstruction via 64 slices covering $1 \times 1 \times .8mm$ region, the partial data is sampled via the trajectory that distributed along the radial lines in each slice.

4.2.3 Conclusions

We have presented the details of the implementation of the proposed algorithm ALSR for sparse optimization as well as its application for rapid MR imaging. We presented the numerical experiments for reconstructing 2D and 3D MR images via ALSR, sparseMRI and RecPF, and the results show present that all three packages can exploit the sparsity of the images and reduce the scanning time significantly via undersampling in k-space. The brain imaging experiments demonstrate that the imaging speed of ALSR and RecPF are comparable and these two are much faster than sparseMRI.

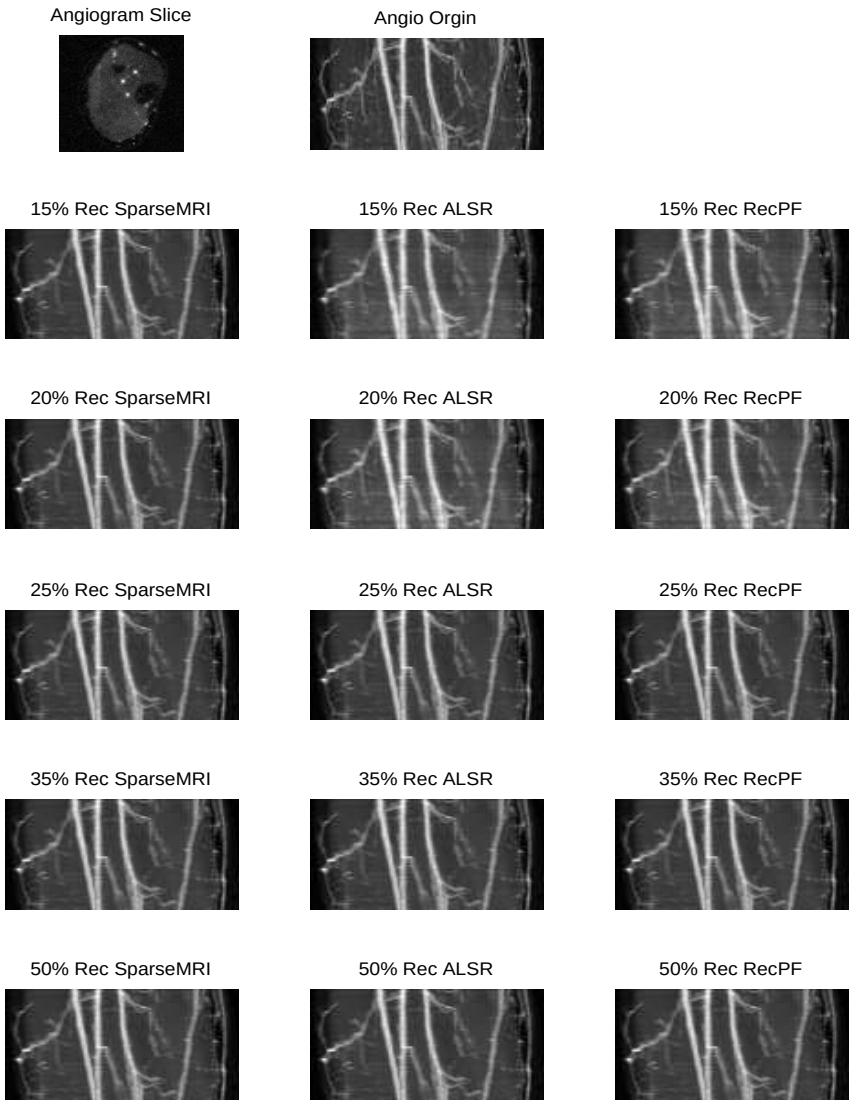

Figure 4.11: Figure depicts an Angiograms of leg, reconstructed via the packages ALSR, sparseMRI and RecPF. We test the behavior the packages under various undersampling rates. The first row is full Nyquist rate image of a slice and the whole scanned section. The images from the second row to the last rows are the 3D angiograms reconstructed via undersampling in k-space. The sparseMRI, ALSR and RecPF enhanced the contrast, and as more sample are involved in the reconstruction, more tiny vessels appear. This shows all three packages can enhance the contrast while preserving the sparsity very well. But the imaging speed time varies a lot to reconstruct the images in the last row, sparseMRI takes 96.2s, ALSR 67.1s, RecPF 65.2s.

Besides, the undersampling in k-space will not hurt the quality of the images when the conditions of CS are satisfied.

Chapter 5

Source Localization Detection with Sparse Reconstruction

In this chapter, we present a general framework for the source localization detection using sparse reconstruction. We first introduce the mechanics of the uniform linear array (ULA) and the relative waveform. Next we develop a well designed over-complete basis so that the problem can be reformulated as an inverse problem with sparse underlying variables. An efficient and robust algorithm using sparsity enforcing regularization for both single time and joint time observations are proposed in the following. We will carry out a series of numerical experiments to show how the proposed algorithm works. Finally we compare the proposed algorithm with some existing packages. At the same time some practical issues of the source localization will be covered.

5.1 Introduction to source localization detection and some existing non-parametric methods

A major application of sensor arrays is the estimation of parameters of the impinging signal. Parameters to be identified include number of signals, magnitudes, frequencies, direction of arrival (DOA), distances and speeds of the signals. The source localization detection has been active and playing a fundamental role in signal processing and the DOA detection arises in many applications, including spectral estimation, signal reconstruction, signal classification and tomography. In this paper, we mainly focus on the detection of DOA [MA05] [IB97] [J.J01] for the narrow band signal in the far field with the uniform linear array. In this case, the wavefront formed by the signal can be treated as planar, that is the distance is irrelevant, and the relative simple array geometries can easily pose the array signal representation as sparse [DD93] [S.N01].

The location of a point in three dimensional space is defined by range, azimuth and elevation. The range is often measured by the return time of travel in active systems

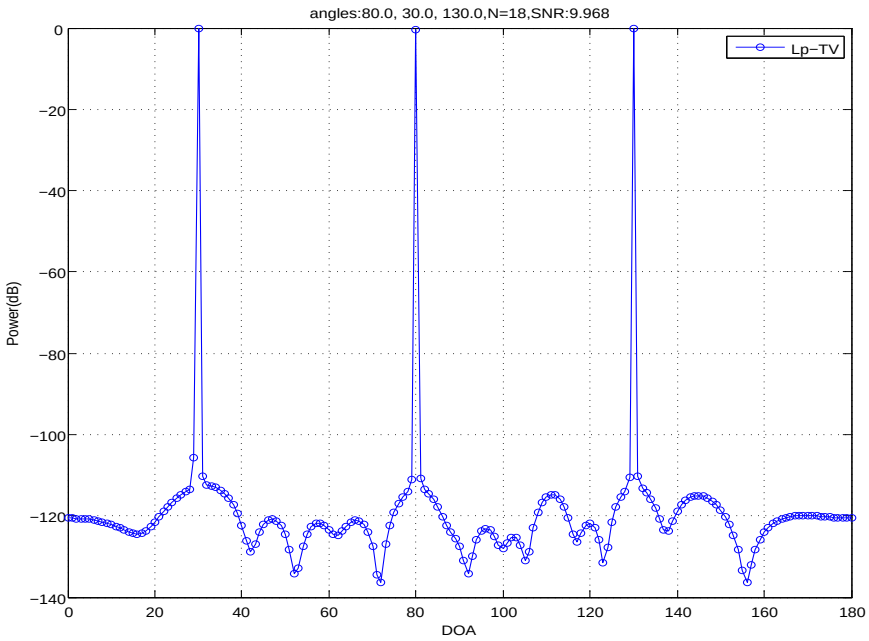

Figure 5.1: The result of Source localization detection with Lp-TV regularization based on 200 joint time samples. Spatial spectra of three sources with DOA's of $30°, 80°, 130°$ and SNR = 12dB

and the relative time of delay among a number of sensors. The azimuth and elevation are obtained from the measurements of DOA by the sensor array [S.N01]. So the DOA detection plays an important roles in signal processing. The goal of source localization is to detect the DOA of wave-fields that impinge on an array consisting of a number of sensors. This task can be approached by sampling the spatial and temporal wave-field, which includes the variation of the time evolution of the sources' energy locations. Here we use the sensor array composed of multiple sensors instead of a single sensor this mainly because the array can bring an apparent improvement in the signal to noise ratio (SNR), the possibility of electronic steering and the robustness of the estimations. After the required information is collected from the array, we can form an appropriate mathematical model and detect the DOA of the impinging signal by solving this model by an appropriate numerical method.

Generally speaking, the DOA estimation methods can be classified into two

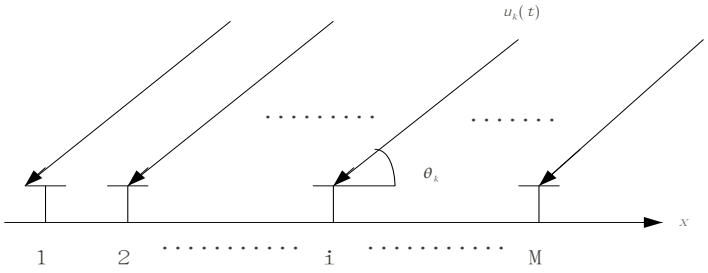

Figure 5.2: The uniform linear array consists of M equal spaced sensor with K impinging narrowband signals. Our goal is to detect the unknown impinging degree of arrivals (DOA) of the narrowband signals and the unknown number of sources K via the M sensor output corrupted with strong background noise.

main categories, namely spectral-based (Non-parametric) approach and Parametric approach. The principle of the parametric methods is to maximize the power of the beam-forming output for a certain given input signal. Different power definitions result in different spectral-based algorithms, such as Capon [J.C69] and MUSIC [R.O81]. The parametric methods, which are known as maximum likelihood (ML) methods, are based on the selection of the likelihood function obtained from the different models to be estimated. The deterministic ML algorithm assumes that the signal waveform is deterministic but unknown, while the stochastic ML algorithms assumes that the signal waveform is a Gaussian random process. In our work, we follow a different approach by posing the array signal representation as sparse under an overcomplete basis, and complete the detection by solving a regularized inverse problem.

The approach to DOA detection by exploiting the sparsity of the underlying signal under a specific overcomplete basis, is made by taking advantage of the geometry of the ULA. We define the signal impinging on the array as zero in all sampling grid points except at the grid points where the energy is detected by the sensor array. In this way, we actually pose the problem as the estimation of a sparse vector with $0-1$ entries under a specific atom of the limited data collected by the sensor array. This indicates

that it is possible to reformulate the DOA detection problems as an inverse problem with sparse underlying variables, and the underlying unknowns can be reconstructed with the aid of regularization.

Using the sparse reconstruction [S.S99] and Compressive Sensing (CS) [D.L06] [E.C06] to improve the estimation performance and robustness in sensor array processing with presence of noise, are gaining more and more popularity. A signal is sparse when it contains a small number of nonzero components, that is the ℓ_0-norm of the underlying variables is minimized. But minimizing the number of nonzero leads to a combinatorial problem that is NP-hard. It is well known that the ℓ_1-norm minimization is an ideal alternative approach [S.S99] [WY08] to enforce the sparsity and is more tractable computationally [E.C06] [E.C04] [EY07]. We found that under certain conditions, the ℓ_p-norm, where $0 < p < 1$, shows even better properties in restoring the sharp features and can beat the ℓ_1 norm in the source localization application. A detailed discussion on the DOA detection using the sparsity regularization will be given in the following sections.

5.2 The uniform linear sensor array and waveform

The uniform linear array (ULA) is one of the most commonly used array geometries in large military phased array systems, such as sonars and radars. A wavefront propagating across the array is captured by the sensors, and each sensor can make an output which is simply a delayed replica of the original waveforms. An array signal is formed and the outputs can be combined in some optimal manner so that the coherent signal emitted by the source is received and other additional inputs are discarded as much as possible.

Consider a ULA consisting of M sensors placed on an equispaced linear grid along the x-axis with distance d to each other. Let $f_m(t)$, $m = 0, 1, 2, \cdots, M-1$ denote the outputs of the m-th sensor, and assume that the signal arrives at successive sensors

with an incremental delay. Suppose the output of the first sensor is $f_0(t) = f(t)$. Then the output of the m-th sensor is $f_m(t) = f(t - m \triangle t)$, where $\triangle t$ denotes the relative delay in each sensor. The m-th sensor output in the frequency domain can be obtained via the Fourier transform:

$$f_m(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{j\omega(t - m\triangle t)} d\omega, \tag{5.1}$$

where the frequency representation $\hat{f}(\omega)$ is given by the inverse Fourier transform

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(u) e^{-j\omega u} du.$$

Traditionally, the direction of arrival (DOA) of the impinging signal is measured with respect to the normal to array aperture, and denoted by θ. Suppose the frequency of a wavefront propagating in a certain medium is ω, and the relative speed is c, then the delay time $\triangle t$ between consecutive sensors can be represented as $\triangle t = \frac{d}{c} \sin\theta$. Let $w_0, w_1, \cdots, w_{M-1}$ be a set of weight coefficients for the beamformation. The beam outputs of the array in terms of the weighted sum of each sensor output is given by

$$f(t) = \sum_{i=0}^{M-1} w_i f_i(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{j\omega t} d\omega \sum_{i=0}^{M-1} w_i e^{-j\omega m \frac{d \sin\theta}{c}}, \tag{5.2}$$

where $R(\theta) \triangleq \sum_{i=0}^{M-1} w_i e^{-j\omega m \frac{d \sin\theta}{c}}$ is the so called array response function defined only by the nature of the ULA. As we know, a narrow-band signal $f_{nb}(t)$ can be represented as

$$f_{nb}(t) = A(t) \cos(\omega t + \varphi(t)), \tag{5.3}$$

where the envelope $A(t)$ and phase $\varphi(t)$ are slowly varying. $\cos(\omega t)$ is a rapidly varying sinusoid (here we only take the $\cos(\omega t)$ case as an example, actually it also can be $\sin(\omega t)$ or both) with carrier frequency ω. Then the narrow-band signal $f_{nb}(t)$ in (5.3) can be defined as:

$$f_{nb}(t) = f_i(t) \cos(\varphi(t)) - f_q(t) \sin(\varphi(t))$$

where $f_i(t) = A(t) \cos(\omega t)$ and $f_q(t) = A(t) \sin(\omega t)$ are the inphase and quadrature components, respectively. Suppose a general complex analytical signal is denoted as

$f_c(t) = f_i(t) + jf_q(t)$. Then a narrow-band signal delayed by a quarter period $\frac{T}{4}$ can be represented as the sum of inphase and quadrature components as follows:

$$\begin{aligned} f_{nb}(t-\frac{T}{4}) &= f_i(t)\cos(\omega(t-\frac{T}{4})) - f_q(t)\sin(\omega(t-\frac{T}{4})) \\ &= f_i(t)\sin(\omega t) + f_q(t)\cos(\omega t). \end{aligned} \tag{5.4}$$

Hence a complex analytical narrow-band signal can be expressed through a process referred to as quadrature filtering:

$$\begin{aligned} f_{nb}(t) + jf_{nb}(t-\frac{T}{4}) &= f_i(t) + f_q(t) + j(f_i(t)\sin(\omega t) + f_q(t)\cos(\omega t)) \\ &= f_i(t)e^{j\omega t} + jf_q(t)e^{j\omega t} \\ &= f_c(t)e^{j\omega t}, \end{aligned}$$

and the relative output of the ULA shown in (5.2) for the narrow-band case can be expressed in a matrix format.

Suppose a beam output of the m-th sensor is represented via a quadrature filter as:

$$\begin{aligned} f_m(t) &= f_{nb}(t - m\frac{d\sin\theta}{c}) + jf_{nb}(t - m\frac{d\sin\theta}{c} - \frac{T}{4}) \\ &= f_c(t)e^{j\omega(t - m\frac{d\sin\theta}{c})}, \end{aligned}$$

In this case, the relative time delay in the m-th sensor $\{t - m\frac{d\sin\theta}{c}\}$ is measured with respect to the distance from the first sensor to the m-th sensor and appears in the complex sinusoid. If the the output of the first sensor is $f_0(t) = f_c(t)$, then the phase correction of the initial sensor $j\omega t$ in each of the sensor can be dropped and the relative m-th sensor response with respect to the initial sensor can be simplified as

$$e^{-j\omega m\frac{d\sin\theta}{c}}, m = 0, 1, \cdots, M-1.$$

Therefore, the output of the m-th sensor for the narrow-band single source is given by:

$$f_m(t) = f_c(t)e^{-j\omega m\frac{d\sin\theta}{c}}, \quad where \quad m = 0, 1, \cdots, M-1, \tag{5.5}$$

Let us consider a snapshot vector $f^T(t) =< f_0(t), f_1(t), \cdots, f_{M-1}(t) >$. Each of its component $f_i(t)$ represents the output of each sensor taken at the same time instance t. If there are N narrow-band sources radiating simultaneously, then the array output can be expressed as a linear combination of the type (5.5) as follows:

$$f(t) = \sum_{m=0}^{M-1} f_m(t) = \sum_{m=0}^{M-1} f_{c_i}(t) e^{-j\omega m \frac{d \sin\theta}{c}}, \quad i = 0, 1, \cdots, N-1, \tag{5.6}$$

where $f_c^T(t) =< f_{c_0}(t), f_{c_1}(t), \cdots, f_{c_{(N-1)}}(t) >$ is a complex signal associated with the N narrow-band sources. Mathematically, we can rewrite the array representation (5.6) into a compact format:

$$f(t) = a(\theta_0) f_c(t) \tag{5.7}$$

where $a(\theta_0) = e^{-j\omega m \frac{d \sin\theta_0}{c}}, m = 0, 1, \cdots, M-1$ is the array response representing the propagation effect of the medium on a wavefront across the array. Generally we can rewrite equation (5.7) into a more compact form as:

$$y(t) = A(\theta) u(t) + N(t) \tag{5.8}$$

where $A(\theta) = [a(\theta_0), a(\theta_1), \cdots, a(\theta_{N-1})]$ is the array response matrix representing the underlying DOA of the impinging narrow-band signal, and each of its column $a(\theta_i)$ is referred as the steering vector which steers the array to the direction θ_i, $y(t) \in C^{M \times 1}$ is a snapshot of the array output and used as the observation, $N(t) \in C^{N \times 1}$ is the additional noise and $u(t) \in N \times 1$ is an underlying complex signal reflecting the the source among the N possible directions.

It is worth to point out that in (5.8) $\theta = [\theta_0, \theta_1, \cdots, \theta_{N-1}]$ is an unknown signal parameter reflecting the DOA of N narrow-band sources, and each component of the vector $s(t)$ reflects the signal in the direction pointing to θ. However, for the broadband source we have to work in the frequency domain instead of the temporal domain and the array output can be represented as a function $f_m(\omega, \theta)$ (5.1) via Fourier transform, and (5.8) becomes $y(\omega) = A(\omega, \theta) s(\omega) + N(\omega)$. In this work we mainly focus on the

development of the DOA estimation method in the narrow-band case and the model (5.8) plays a core role in DOA estimation.

5.3 The overcomplete basis and sparse signal representation

In last section, we reformulated the array output of N narrow-band signals into a compact matrix format (5.8). In this section, we will modify the setting of the ULA and form an overcomplete basis by redefining some parameters of the array and the impinging signals, such that the underlying signal coming from the N sources has a transformed sparsity under this basis. We will also deal with the joint time problem, which is a natural generalization of the source localization problem, to process the multiple measurements in the temporal domain.

We notice that the manifold matrix $A(\theta)$ in equation (5.8) is parameterized by the DOA, which is going to be determined by the measurements collected from the ULA, and the relative complex variable $u(t)$ reflects the signal coming from the specific direction indexed with θ_is. Here we can consider to adapt the definitions of parameters in model (5.8) to make $u(t)$ sparse under a deterministic under-determined basis, and finally get an inverse problem as well as an efficient numerical solution that can lead to a robust estimation of the degree of arrival.

Traditionally, the degree of arrival θ is measured with respect to the normal to array aperture, and it uses the first sensor along the positive x-axis as the phase center. Then the associate output delay from the following consecutive sensors can be expressed by $\triangle t = \frac{d}{c}\sin\theta$, where $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Now suppose the aperture of the array can receive the impinging signal from all directions it faces to, then we can change the definition of the impinging angle θ and redefine it as the angle from the array aperture along the positive x-axis to the impinging signal. In this way, the range of θ becomes $[0, \pi]$ and the relative delay of each consecutive sensor becomes $\triangle t = \frac{d}{c}\cos\theta$. Let us take $\tilde{\theta} = \{\tilde{\theta}_0, \tilde{\theta}_1, \cdots, \tilde{\theta}_{N-1}\}$ as all possible source locations that the

aperture of the sensor can cover, and use each $\tilde{\theta}_i$ as grid points from which sensors in ULA collect the data. Then the signal $u(t)$ impinging on the aperture are not zero at a few griding points $\tilde{\theta}_i$ when the true DOA coincident the griding point, otherwise it is zero. Since the number of the sources is small compared with the number of gird points N, this indicates that $u(t)$ has only a few nonzero components comparing with the total number of the sampling points. Hence, in this way, the sensing matrix $A(\tilde{\theta}) = [a(\tilde{\theta}_0), \cdots, a(\tilde{\theta}_{N-1})]$ becomes deterministic, denoted by A, and each it's column $a(\tilde{\theta}_i)$ are the so called steering vectors reflecting the signal steered in the direction θ_i. Represented under this basis, the underlying $N \times 1$ signal field, denoted by $u(t)$, is k-sparse. Therefore the single time model (one snapshot vector indexed by time t) can be expressed as:

$$y(t) = A(\tilde{\theta})u(t) + n, \tag{5.9}$$

where the $N \times 1$ vector $u(t)$ is the underlying variable which is sparsely represented under the deterministic manifold matrix $A(\tilde{\theta}) \in C^{M \times N}$. It needs to be pointed out that M is the number of sensors in ULA and N the number of the grid points which is naturally much greater than M. Then the steering vectors in $A(\theta_i)$ form an overcomplete basis. In practical implementations, we can shift the phase center to the midpoint of the ULA such that the sensor outputs are symmetric at the both sides of the phase center, and the relative distance from the pth sensor to the phase center is $(p - \frac{M-1}{2})d$, the associate time delay of the consecutive sensor are measured in terms of the distance from the each sensor to the phase center. In this setting, we can rewrite the relative sensing matrix $A(\tilde{\theta})$ in (5.9) as:

$$\begin{aligned} A(\tilde{\theta})_{pq} = a(\tilde{\theta}_q)_p = & e^{-j\omega(p-\frac{M-1}{2})\frac{d}{c}\cos\tilde{\theta}_q}, \\ \textit{where} \quad & p = 0, 1, \cdots, M-1, \\ & q = 0, 1. \cdots, N-1. \end{aligned} \tag{5.10}$$

Here it is worth to point out that, since the sensor position is symmetric to the array

center, the associated sensing matrix $A(\tilde{\theta})$ in (5.10) is hermitian, that is $A^H = A$, and it's eigenvalues $\lambda(A)$ are real valued.

The model (5.9) treats the underlying signal energy as the function of the hypothesized source location, and the energy spectrum of the underlying signal $u(t)$ is sparse. A similar philosophy of transforming the underlying signal into a sparse representation under a deterministic overcomplete basis to estimate the signal parameters is presented in [MA05][IB97][J.J96][S.S99]. This application is application was getting more and more popular in sensor array processing. In our work, we mainly focus on developing numerical algorithms based on the sparse reconstruction techniques and compressive sensing for solving the inverse problem (5.9) for both the single time data (when $T = 1$) and the multiple time data (when $T \geq 1$).

5.4 The inverse problem with multiple measurement vectors and its numerical solution

We formulate the DOA detection model as a classic linear inverse problem (5.9) with sparse underlying unknowns. Although the single snapshot processing may have its own applications, usually in sensor array processing the multiple snapshot observation data is of more practical importance. In this case, the multiple time measurement

$$Y = [y(t_0), y(t_1), \cdots, y(t_{T-1})]$$

collected by the sensor array is a time series of the impinging signals, and the components in each column $y(t_i) \in C^{M\times 1}$ are the spectrum in each grid point in the spatial domain, the data in each row are the outputs of the associated sensor with respect to the moments $t_0, t_1, \cdots, t_{T-1}$ of collecting the data. Mathematically, the model of DOA estimation with multiple measurement can be expressed as:

$$Y = AS + N, \tag{5.11}$$

where the deterministic $M \times N$ steering matrix A determined by the physics of the array is the same as the one defined in (5.9), but the measurement $Y = \{y(t_0), \cdots, y(t_{T-1})\}$ is the observation of multiple snapshots taken at each moment $\{t_0, t_1, \cdots, t_{T-1}\}$, and $N \in C^{M \times T}$ is the additive Gaussian white noise whose elements in each column are the noise in the snapshot of the relative moment. In this multiple snapshot problem, the underlying unknowns $S \in C^{N \times T}$ become 2-D and reflect the impinging signal from the DOA $\tilde{\theta}_i$ at each time t_i. Our main goal in this section is to exploit the numerical solution of the source localization with joint time model (5.11) through the idea of sparsity enforcing regularization and compressive sensing.

5.4.1 The joint measurement model and compressive sensing

Naturally we may think of treating each time index $t_i \in \{1, 2, \cdots, T\}$ separately and transform the joint time model (5.11) into T single time models (5.9). Then we would have a set of T solutions $\{\hat{s}(t_i) | y(t_i) = A\hat{s}(t_i) + n(t_i), i = 0, \cdots, t_{T-1}\}$ by solving each of T single time models. Using all the information brought by the solution of each single time model $\hat{s}(t_i)$ and remove the redundancy in temporal domain by statistic methods or other ways to form an estimator representing the unique estimation on the source locations of the impinging signal. Usually there are several ways to reduce the redundancy and form an efficient estimation on the source locations from the T solutions $\hat{s}(t_i)$, such as, taking the mean and find out the peaks, using cluster analysis or some other ways in statistics. Apparently treating each time index separately is not practical, inefficient and not robust numerically. Especially, when the observation data has high noise level or is large scale in the temporal domain, the computation load of this scheme is dominated by the cost of solving each T inverse problems and is linearly proportional to the dimension in temporal domain; on the other hand, since the data collected at different time t_i are processed separately, the final estimation will become highly sensitive to the additional noise in the observation $y(t_i)$. Alternatively,

we may consider to reduce the redundancy of data Y in the temporal domain first, and then find out a robust way of estimating the source locations from (5.9) while avoid processing the data collected in different time separately. In this work we propose a simple way of reducing the redundancy of the joint time observation, leading to a new robust estimation on the source locations utilizing *Compressive Sensing* (CS) and the application of the sparsity enforcing regularization.

CS [D.L06][E.C06][E.C04] is a way of reconstructing an underlying unknown which has potential sparsity or transformed sparsity under a overcomplete basis. The whole process of the CS consist mainly of the encoding and decoding steps. In the encoding step, let the vector u denote a signal of interest. Ψ denotes a known sparsifying basis such that $u = \Psi s$ has a sparse representation under Ψ. Here sparse means that there are only a small number of nonzero entries in s and the others are zero. Then we can solve for s by minimizing a ℓ_1-norm related problem in the decoding process such that the underlying unknown u can be reconstructed by using the measurement y of a linear projection of u onto Φ, that is the linear measurement y is obtained from $y = \Phi u = \Phi\Psi s$, and usually y is the partial sampling of the whole underlying object. The main result of CS states that when the matrix $\Phi\Psi$ possesses the of *restricted isometric property* (RIP) [E.C06], then u can be reconstructed exactly with high probability by solving an l_1-norm related linear programming

$$\begin{aligned} \min_{s} \quad & ||s||_{\ell_1} \\ s.t. \quad & y = As, \end{aligned} \tag{5.12}$$

where $A = \Phi\Psi$, (5.12) is known as the basis pursuit [S.S99]. The model (5.12) can be easily extended to the joint time case: the relative unknowns becomes $S \in C^{N\times T}$, which is a matrix with each row representing the estimation of the spectrum at one grid point with respect to all the moments, and each column of S representing the estimation of the source location for the specific one snapshot. So we may pursuit the sparsity of S along

each of its columns, but the row vector reflecting the spectrum in the temporal domain is not sparse necessarily. Let $s_i = \sum_j ||S_{ij}||_2^2$ denote the Euclidean norm of the i-th row of S. This leads to a scheme of minimizing $||s||_{\ell_1} = \sum_i |s_i| = \sum_i \sqrt{\sum_j ||S_{ij}||_2^2}$ such that the error is minimized. Mathematically, we can write this joint sparse reconstruction model as:

$$\min_S \sum_i \sqrt{\sum_j ||S_{ij}||_2^2} \tag{5.13}$$
$$s.t. \quad Y = AS.$$

The joint sparsity reconstruction model and related numerical algorithms are discussed in [JX06][E.B09][J.J04][MY08][SKD05]. In our work, we will try to use a different way to reduce the redundancy of the data in the temporal domain and a different functional to detect the source localization.

The source localization problem can be formulated into a problem with sparse underlying unknowns represented under the adapted overcomplete basis (5.10), the CS is suitable for solving this problem. Since $A \in C^{M\times N}$ in (5.12) is determined by the response of the ULA as well as the physics of the sensor array, the sensing matrix A does not change as the dimensionality in temporal domain increases. So for the stationary source, both the single time and joint time measurement problems have the same sensing matrix. Our goal is to find out a sparse vector representing an aggregate estimation of the source locations of the impinging signals such that the error of the detection is minimized based on the given joint time measurement Y and the array response. In this case, all the snapshots $\{y(t_i), i = 0, \cdots T-1\}$ imply the same spectrum distribution in the spatial domain. Then instead of solving the joint measurement problem (5.16), we can consider to minimize the number of nonzeros of the spectrum of the source locations, while minimizing the level of the mean square error of the estimated source locations regarding to the measurement of each snapshot

$$MSE = \frac{1}{T}\sum_{i=0}^{T-1} ||As - y(t_i)||_2^2.$$

This approach is more robust with respect to noise, more economical in the computational effort compared with the model (5.16) and more suited to the physics of the source localization problem. Let the $N \times 1$ vector s denote the spectrum of the source in the spatial domain. Then the sparse joint measurement model (5.16) can be rewritten as:

$$\begin{aligned} \min_{s} \quad & H_{reg}(s) \\ s.t. \quad & \frac{1}{T}\sum_{i=0}^{T-1} ||As - y(t_i)||_2^2 \leq \varepsilon, \end{aligned} \tag{5.14}$$

where $\varepsilon \to 0$ is a small positive value controlling the noise level, the functional $H_{reg}(\cdot)$ is the regularization determined by the prior knowledge of the underlying objective, and in practice, it is not limited to the ℓ_1-norm. Other regularization, such as Total Variation or ℓ_p-norm may even perform better in some circumstances. We need to point out that the dimension of the underlying variable s in (5.14) does not increase with increasing measurements in the temporal domain, but the dimension of the underlying variable S in (5.16) will increase as more data in the temporal domain get involved. On the other hand, the model (5.14) controls the noise level by using the mean square error of the sample from each time t_i with equal weight as the constraint, and the relative computational load does not increase with increasing numbers of temporal samples. Besides, using different regularization or their combinations $H_{reg}(s)$ as the objective function leaves more freedom to the user, and the specific regularization can be determined based on the prior knowledge of the underlying object, and the relative computational issues for dealing with various regularization also raised at the same time. Therefore, an efficient and robust numerical algorithm is needed, and in the following section, we mainly focus on the development of numerical methods for solving model (5.14) with respect to various conditions of the DOA detection problem.

5.4.2 The regularization

We want to find a sparse solution satisfying the constraint condition by solving the model (5.14) with the regularization $H_{reg}(s)$, which is a functional of the underlying variable. Naturally, minimizing the ℓ_0-norm of s may be a possible way to pursue the optimal solution.The ℓ_0-norm is defined as:

$$||s||_0 \triangleq \lim_{p\to 0} \sum_i |s_i|^p,$$

however, using the ℓ_0 regularization to minimize the number of nonzero entries is an integer program and this will lead to a NP-hard problem [B.K95][MD79], its numerical solution is practically untractable. Alternatively, we may consider to use other norms which permit a reliable numerical solution in place of the ℓ_0-norm. The ℓ_1-norm (when $p = 1$) has been widely used as an alternative of the ℓ_0-norm for its convexity and tractability in computation. Many algorithms for solving the ℓ_1-norm regularized inverse problem have been proposed, such as [dBM07][EY07][J.A06][JA05][MS07b] [MS07a][RM01] *.etc*.

From the definition of the ℓ_p-norm, we notice that as p approaches 0 the optimal solution becomes the sparsest. Although the ℓ_1-norm approximation performs better in numerical computation, generally the solution derived from the ℓ_1-norm regularization is not as sparse as in the ℓ_0 case. Here we may think of a norm with $0 < p < 1$ in the hope that this ℓ_p-norm can provide a solution that is sparser than the ℓ_1-norm case and more tractable in computation than for the ℓ_0-norm. Actually, it is reasonable to expect the solution derived from the ℓ_p-norm to be sparser, since it is a tighter approximation to the ℓ_0 norm. It must be pointed out that ℓ_p-norm is not a true Euclidean norm since the triangle inequality is not satisfied. The ℓ_p-norm regularization is non-convex, and we can only expect to obtain a local optimum, instead of the global one. But according to our numerical experiments, the non-convexity is not an obstacle in our

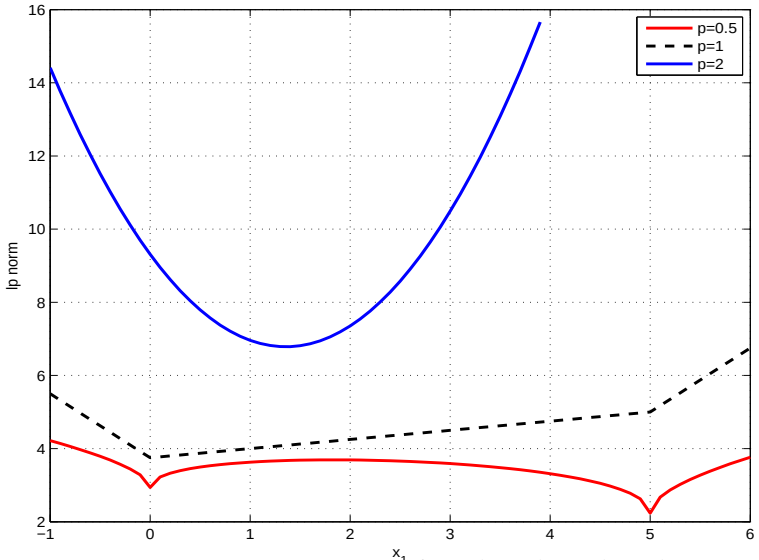

Figure 5.3: The optimal solution $\hat{x} = (x_1, 3 - \frac{3}{5}x_1, \frac{1}{2} - \frac{1}{10}x_1, \frac{1}{4} - \frac{1}{20}x_1)^T$ of the underdetermined linear system (5.12) under the l^p norm regularization. When $p = 2$, the $\hat{x}$ is not sparse; when $p = 1$ the solution is sparse; but when $p = 0.5$ the solution $\hat{x}$ is even sparser than the case $p = 1$.

source localization problem.

Next, we give an intuitive example to state how ℓ_p-norm performs in exploiting the sparsity of the solution of a full row rank linear system. By comparing with other norms ($p = 1, 2$) we can easily see that the l^p-norm implies a sparser solution. Let us consider an underdetermined linear system:

$$\begin{pmatrix} 0.4 & \frac{2}{3} & 0 & 0 \\ 0.4 & 0 & 4 & 0 \\ 0.4 & 0 & 0 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \tag{5.15}$$

and its solution:

$$\hat{x} = (x_1, 3 - \frac{3}{5}x_1, \frac{1}{2} - \frac{1}{10}x_1, \frac{1}{4} - \frac{1}{20}x_1)^T.$$

We notice that the solution of this system is not unique, and when $x_1 = 5$, the solution

$\hat{x} = (5,0,0,0)^T$ attains the sparsest form. Let us define the p-th power of the ℓ_p-norm of the solution to this linear system (5.15) as:

$$||\hat{x}||_p^p \triangleq |x_1|^p + |3 - \frac{3}{5}x_1|^p + |\frac{1}{2} - \frac{1}{10}x_1|^p + |\frac{1}{4} - \frac{1}{20}x_1|^p. \tag{5.16}$$

We can try to choose various p values to compare the sparsity of the solution regularized under each norm. When $p = 2$, the minimum energy of this linear system is reached at $x_1 = 1.357$, and the relative ℓ_2-norm regularized solution is

$$\hat{x}_{\ell_2} = (1.357, 2.1858, 0.3643, 0.2432)^T,$$

apparently this is not sparse. Figure 5.3 depicts the solutions of the linear system (5.15) regularized by various norms and from it we notice that: when $p = 1$, the optimal solution is of the linear system is reached at $x_1 = 0$, and the relative solution

$$\hat{x}_{\ell_1} = (0, 3, 0.5, 0.25)^T,$$

is sparser than $\hat{x}_{l^2}$; if we set $p = 0.5$, the ℓ_p-norm regularized solution of the solution of this linear system is reached at $x_1 = 5$, and the relative solution

$$\hat{x}_{\ell_p} = (5,0,0,0)^T,$$

is the sparsest one, which is the sparsest case according to theory. As a closer approximation to the ℓ_0-norm, ℓ_p-norm regularization shows a better properties on exploiting the sparsity than the ℓ_1-norm.

Another regularization of our interest is the *Total Variation* [LE92], which has been widely used in many areas of image sciences for its nice properties of enhancing sharp edges and restoring discontinuities. Let $u(x,y)$ denote the observed intensity function of a pixel value in a noisy image, where $x, y \in \Omega$. The TV-norm of this image can be expressed as:

$$TV(u) = \int_\Omega \sqrt{u_x^2 + u_y^2} dxdy, \tag{5.17}$$

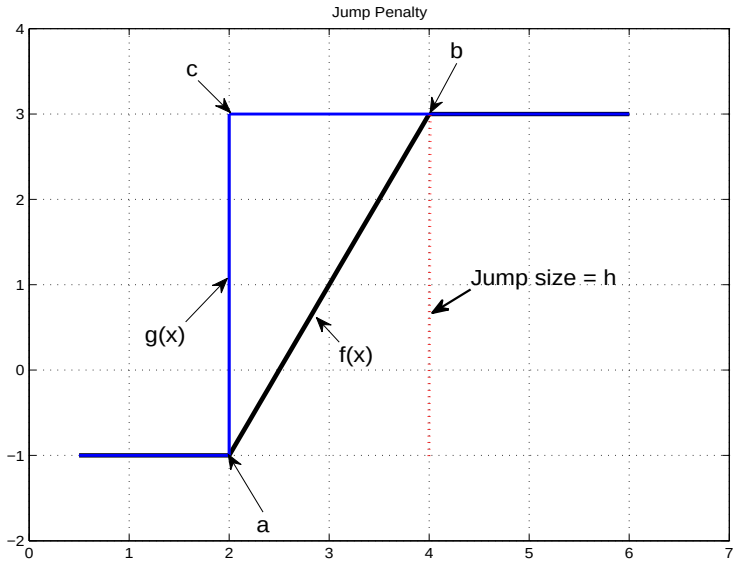

Figure 5.4: The TV norm and the magnitude of the jump

and its related discretized version is given by

$$TV(u) = \sum_{i,j} \sqrt{u_x^2(x_i, y_j) + u_y^2(x_i, y_j)} \tag{5.18}$$

where $(x_i, y_j) \in \Omega$ are the grid points in the image domain. In signal precessing, the TV-norm of the 1-D signal is equivalent to the l^1-norm of the finite difference of the underlying variables along the grid points:

$$TV(s) = \sum_{i \in \Omega} |s_{i+1} - s_i|, \tag{5.19}$$

in this way, the magnitude of the jump is not penalized by the denoising. The Figure (5.4) shows how the TV-norm works on preserving the jump: let $f(x)$ and $g(x)$ represent two piecewise smooth functions with $g(x)$ having a sharper discontinuity. Then the value of the TV-norm of the function f is equal to the magnitude of the jump,

$$TV_f = \int_a^b |\frac{h}{b-a}| dx = |h|. \tag{5.20}$$

As the jump size $g(x)$ gets sharper as a approaching b, the limit value of this TV-norm approaches the jump size of the function $f(x)$,

$$TV_g = \lim_{\varepsilon \to 0} \int_a^{a+\varepsilon} |\frac{h}{\varepsilon}| dx = |h|. \tag{5.21}$$

So the TV-norm preserves the magnitude of the jump while removing the aliasing. Motivated by the comparison of ℓ_1 and ℓ_p norms, we may also consider redefine the TV norm by using the ℓ_p norm and expect to penalize the noise more while preserving more tiny sharp features in the underlying objects. Some of the related work can be found in [Cha07].

In the DOA detection problem, since the number of the sources are sparse compared with the number of all potential ones that we check, and the jump discontinuity with strong background noise also appears, we can consider to use the TV-norm to preserve this block structure. Besides, minimizing the finite differences of sensor response of the consecutive grid points makes the magnitude of the sensor responses between two sources as small as possible, so that the TV-norm is very helpful to identify close distributed sources. Hence based on prior knowledge of the underlying signal we may consider to use a combination of the ℓ_p-norm and TV-norm as the regularization to pursue a desired estimation.

5.4.3 The joint measurement reconstruction algorithm using Lp-TV regularization

In the joint time source localization problem, the observation $Y = \{y(t_1), \cdots, y(t_T)\}$ is a time series of the sample collected from the spatial domain. It is preferable to combine the samples covering up to time T and form one aggregated observation on the source location by taking the benefit of all the samples of the stationary source $y(t_i)$ reflecting the the location of the same sources. In practice, we use the average of Y over the time index t_i as the observation:

$$\bar{Y} = \frac{1}{T} \sum_{i=1}^{T} y(t_i), \tag{5.22}$$

to avoid processing T snapshots individually and the complexity of the problem is reduced greatly. We will show this way is robust and efficient by a number of numerical experiments in the following section. Besides, we know that the underlying DOA is sparse represented under the overcomplete basis (5.10), and both the ℓ_p and TV-norm are well suited for improving the sparsity and enhancing the resolution while removing the noise and preserving the jump magnitude. Therefore we use the weighted sum of the ℓ_p-norm and TV-norm as our objective function and detect the DOA through minimizing this function represented as:

$$\begin{aligned} \min_{s} \quad & w_1||s||_p^p + w_2 TV(s) \\ s.t. \quad & ||As - \bar{Y}||_2^2 \leq \varepsilon, \end{aligned} \tag{5.23}$$

where w_1, w_2 are the weight coefficients of the ℓ_p and TV terms, and these fixed coefficients leave the freedom for the user to customize the parameter values. These parameters can be used to adapt characteristics of specific problems such that make the reconstruction reach an ideal quality. It is known that the problem (5.23) can be converted into an unconstrained problem as:

$$\min_{s} \quad F(s) \triangleq w_1||s||_p^p + w_2 TV(s) + \frac{\lambda}{2}||As - \bar{Y}||_2^2, \tag{5.24}$$

where $0 < p < 1$, λ is a penalty parameter, and we know that as $\lambda \to \infty$ the problem (5.24) is equivalent to (5.23).

The 1D discretized total variation term $TV(s)$ in (5.24) can be rewritten as:

$$TV(s) = \sum_i \sqrt{(D_i s)^2}, \tag{5.25}$$

where $D \in R^{N \times N}$ is the finite difference operator, where $D_i s \triangleq s_i - s_{i-1}$ denotes the forward finite difference on the i-th entries in s, it can also be viewed as the ith row of the matrix D. However, the TV-norm is not suitable for numerical computation because of its non-differentiability. So we can add a slack variable and make the TV-norm a

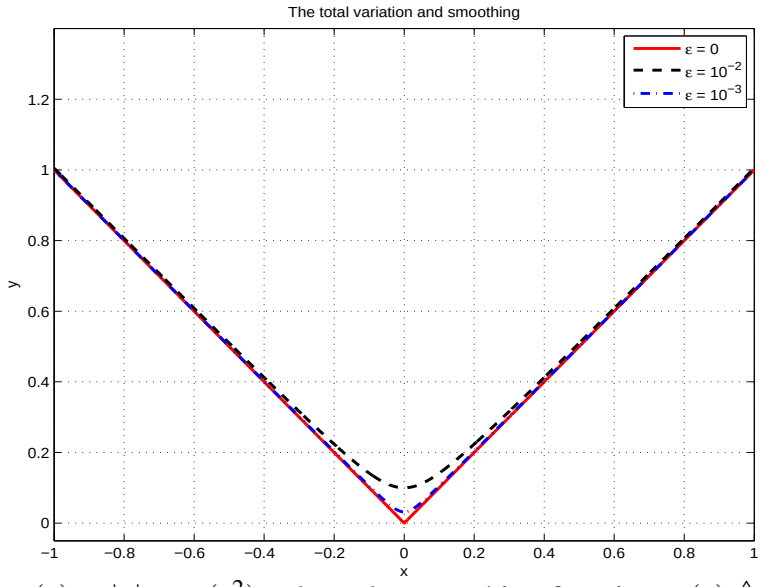

Figure 5.5: $TV(x) = |x| \approx \psi(x^2)$, where the smoothing function $\psi(x) \triangleq \sqrt{x+\varepsilon}$, ε is a constant controlling the smoothness and the approximation accuracy. From this figure we can see that as $\varepsilon \to 0$, the approximated functions are smooth and differentiable all the times, while getting closer and closer to the original one

smooth function as below:

$$\psi(t) = \sqrt{t+\varepsilon},$$

where the constant $\varepsilon \to 0^+$ is the smoothing parameter, and $\psi(t) \approx |t|$. Then in this way, the discretized TV norm can be approximated as:

$$TV(s) \approx \sum_i \psi([D_i s]^2), \tag{5.26}$$

in practice, we can choose the smoothing parameter $\varepsilon \in (10^{-10}, 10^{-5})$. Now the derivative of the approximated TV norm (5.26) with respect to the s_i can be expressed as:

$$\frac{\partial TV(s)}{\partial s_i} = \frac{D_i s}{\psi([D_i s]^2)} - \frac{D_{i+1} s}{\psi([D_{i+1}]^2)}, \tag{5.27}$$

and the gradient of the TV term can be rewritten in a matrix form as:

$$\nabla TV(s) = D^T \Lambda(s) D s. \tag{5.28}$$

where the diagonal matrix $\Lambda(s) \in C^{N\times N}$ is defined as

$$\Lambda(s) \triangleq diag\{\psi^{'}([D_i s]^2)\}, \quad where \quad i = 1, \cdots N.$$

It is worth to point out that the TV-norm can also be implemented by using the l^p-norm as:

$$TV_p(s) = \sum_i ||s_i - s_{i-1}||_p.$$

We can also use a similar technique as we introduced above to remove the non-differentiability in $TV_p(s)$, and since the ℓ_p-norm performs even better in exploiting the sparsity than ℓ_1-norm, we would expect the jump to be even sharper with the aid of $TV_p(s)$.

Next we can in a similar way make the ℓ_p term in (5.24) smooth such that its derivative is approachable. We insert an additional slack variable to smoothen the ℓ_p-norm $(0 < p < 1)$. Mathematically the approximated ℓ_p term can be written as:

$$||s||_p^p \triangleq \sum_i |s_i|^p \approx \sum_i (s_i^2 + \varepsilon)^{\frac{p}{2}}, \tag{5.29}$$

where $\varepsilon \to 0^+$ is the smoothing parameter and the approximated ℓ_p-norm becomes differentiable. Hence, the derivative of the smoothened ℓ_p term in (5.24) is expressed as:

$$\frac{\partial ||s||_p^p}{\partial s_i} \approx \frac{p}{(s_i^2 + \varepsilon)^{1-\frac{p}{2}}} s_i. \tag{5.30}$$

If we define $\lambda_i = \dfrac{p}{((s_i)^2 + \varepsilon)^{1-\frac{p}{2}}}$, then the gradient of the ℓ_p term $(0 < p < 1)$ is expressed in matrix formate as:

$$\nabla ||s||_p^p \triangleq Q_\varepsilon(s) s, \tag{5.31}$$

where the $N \times N$ iteration matrix $Q_\varepsilon(s)$ is diagonal, and defined as

$$Q_\varepsilon(s) = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{pmatrix}.$$

Now we are ready to introduce our algorithm for minimizing the unconstrained problem (5.24). We can find the minimizer of the unconstrained problem by finding the zeros of its gradient. Since the underlying variable $s \in C^{N\times 1}$ can not be isolated from its derivative in both the TV part and ℓ_p part, we can consider to approximate the iteration matrix by using the current value s^k and leaving the separable s as the unknown and solving this system to update s^k and iteratively bring it to the next round of computation till the discrepancy $||s^{k+1} - s^k||$ reaches the stopping tolerance. In this way, the gradient of the TV term at the k-th iteration can be expressed as:

$$\nabla TV(s^k) = D^T \Lambda(s^k) D s^{k+1} \triangleq H_\varepsilon(s^k) s^{k+1}, \tag{5.32}$$

where the iteration matrix is defined as $H_\varepsilon(s^k) \triangleq D^T \Lambda(s^k) D$. Similarly, the underlying variable $s \in C^{N\times 1}$ can be separated from the gradient of the ℓ_p term. The gradient of the fidelity term is expressed as:

$$\begin{aligned} \nabla Fid(s^{k+1}) &= \nabla ||As^{k+1} - \bar{Y}||_2^2 \\ &= A^T A s^{k+1} - A^T \bar{Y}, \end{aligned} \tag{5.33}$$

Now we define the derivative of each part in the objective function (5.23). At the k-th step, the iteration matrix of the gradient of the unconstrained problem (5.24) is given by:

$$\Phi_\varepsilon(s^k) = H_\varepsilon(s^k) + Q_\varepsilon(s^k) + \lambda A^T A, \tag{5.34}$$

and the first order necessary optimality condition of the problem (5.24) is given by:

$$\nabla F(s^{k+1}) = \Phi_\varepsilon(s^k) s^{k+1} - \lambda A^T \bar{Y} = 0. \tag{5.35}$$

So minimizing (5.24) is equivalent to solving the equation (5.35) for s^{k+1} in the k-th iteration. We can summarize our algorithm as below:

It is worth to point out that the linear equation (5.35) can be solved by the conjugate gradient method. We can simply form a diagonal matrix whose diagonal

Algorithm 9 The Framework of Source Localization for the joint time data

Require: $A \in C^{M\times N}, Y = y(t_1), \cdots, y(t_N), s^0 \in C^{N\times 1}, w_1, w_2, \lambda, \varepsilon$

1: Set the parameter values w_1, w_2, λ, ε
2: Set $\bar{Y} = \frac{1}{T}\sum_i y(t_i)$
3: Initialize: $s^0 = A^T\bar{Y}$
4: Set $k \leftarrow 0$
5: **while** not converge **do**
6: Solve equation (5.35): $\Phi_\varepsilon(s^k)s = \lambda A^T\bar{Y}$
7: Set $\{s^{k+1} | \Phi_\varepsilon(s^k)s^{k+1} = \lambda A^T\bar{Y}\}$
8: Update $k \leftarrow k+1$
9: **end while**

are the diagonal entries of the Jacobian matrix $\Phi_\varepsilon(s^k)$, and use it as the preconditioner of the CG method, since this matrix is diagonally dominant and the preconditioned iteration matrix may have eigenvalue close to one. The value of penalty parameter λ can be a fixed appropriate value which ensures the convergence of the routine. In practice a warm start scheme [EY07] for updating the value of this penalty parameter are suggested. Besides, a similar way of linearizing the gradient of the total variation was used by Vogel and Oman in [CM96][CM98] and this method was referred as the lagged diffusive fixed point iteration, and several proofs of the convergence of this algorithms appear in [APL97][DC97][G.A94].

5.5 Implementation and numerical experiment

In this section, we take a series of numerical experiments and presents the numerical results of our proposed ℓ_p-TV method for solving the source localization problem. The discussion covers the regularization parameter selection, the comparison with other algorithms, such as L1-SVD, and some other concerns on the numerical performance of the algorithm such as the robustness of the proposed algorithm with respect to the SNR, number of snapshots *.ect*.

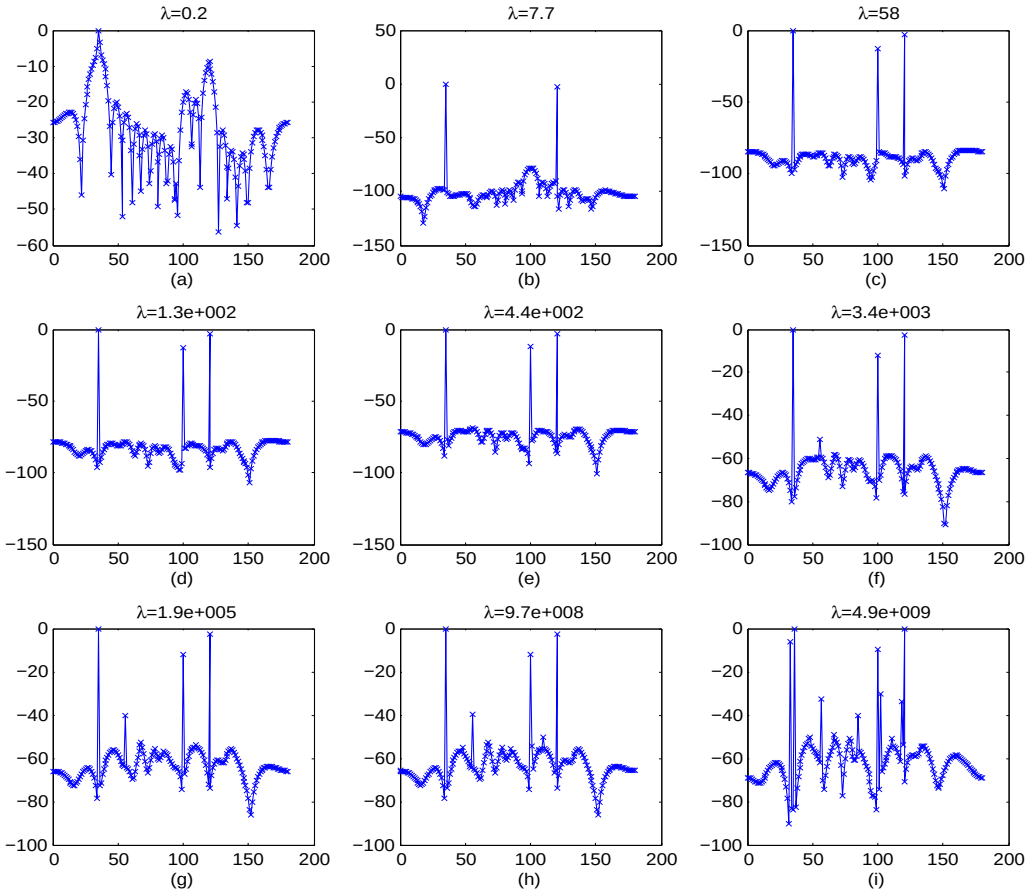

Figure 5.6: From the figure (a)-(i), we compare the detection results under different parameter values and shows how the parameter selection affects the detection result. In this experiment, there are three independent sources $[35^\circ, 100^\circ, 120^\circ]$ and $SNR = 22dB$. When $\lambda = 0.2$ the detection are completely failed since the regulation parameter is too small and the noise is not penalized enough; when $\lambda = 4.9e9$, the true DOA is hardly identified, in this case the parameter is too large and noise is over penalized such that some tiny jump maybe brought by the noise become striking.

5.5.1 Regularization parameter selection

In the unconstrained model (5.24), the regularization parameter λ balances the tradeoff between the sparsity and the fidelity term $||As-\bar{Y}||_2^2$. The choice of this parameter will directly affect the numerical performance of the algorithm. Unfortunately, so far this issue is still an open problem especially for the case when the statistics of the noise is unknown. For the problem with quadratic objective function, such as the Tikhonov regularization, it is possible to express the error of the estimation explicitly in terms of the regularization parameter λ, and the parameter value can be determined by certain optimal principles [A.N63a][AV97][D.L62][R.V02]. However, for the problem with non-quadratic objective function, it is not easy to find out such explicit formula to determine how λ balances the tradeoff. Some well known numerical methods for determining the optimal regularization parameter have been proposed [AD91][MP93][P.C98].

In practice, with too small parameter values the reconstruction is too smooth, but with too large parameter value, the reconstruction shows highly oscillatory artifacts due to noise amplification, as Figure (5.6) shows. If the statistics of the noise is known, a method named discrepancy principle is applicable. The idea of this method is to seek a regularization parameter λ such that

$$\frac{1}{N}||A\hat{s}_\lambda - \bar{Y}||_f^2 \approx E||N||_f^2 = \sigma^2, \tag{5.36}$$

the variance of the estimated error is minimized, where $\hat{s}_\lambda$ is the solution of (5.24) for a given value of λ. Solving the parameter λ from the equation (5.36) requires solving the problem (5.24) for all possible λs, which is rather difficult. On the other hand, if we have no prior knowledge on the statistics of the noise, the choice of the regularization parameter is still impossible via this method.

The L-Curve method [MP93][PD93][P.C92] is a more practical numerical way to determine the optimal value of the regularization parameter. The L-Curve is the

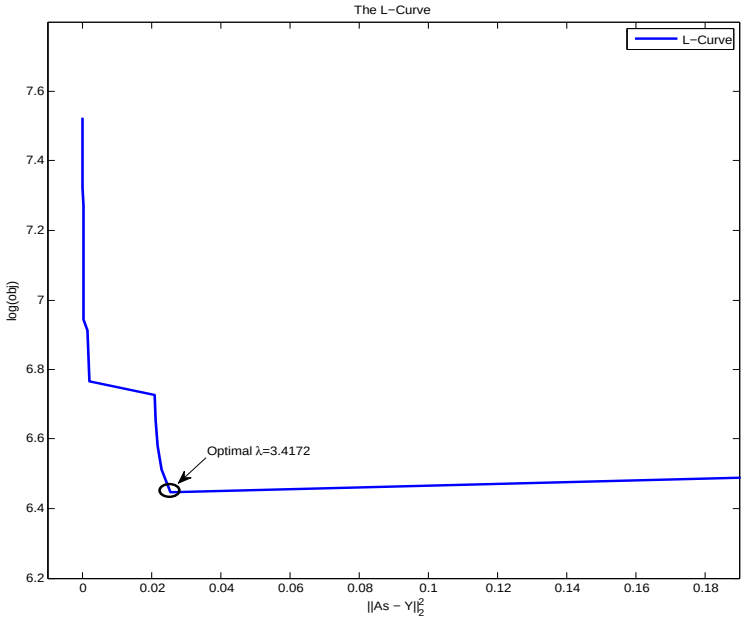

Figure 5.7: It shows that at the corner of the curve, that is, where the regularization parameter $\lambda = 3.4172$, the tradeoff between minimizing the objective function and fitting the fidelity condition is balanced optimally.

plot of the log of the squared norm of the regularized objective function against the square norm of the relative residual for a range of values of the parameter λ. This curve typically has a L shape, and if the parameter λ is too small, the solution $\hat{s}_\lambda$ may not fit the fidelity requirement, but if the value is too large, the algorithm will be expensive and some unexpected and minor features in the noise will be amplified. So the regularization parameter value corresponding to corner of this curve is the one that balances the tradeoff optimally. Besides, this method does not depend on any prior knowledge on the statistics of the noise and more practical.

The optimal choice of the regularization parameter given by the L-Curve method is shown in Figure (5.7). This numerical experiment is based on the number of sensors $M = 18$ and snapshots $T = 200$, and the weight of l^p term is $w_1 = .6$, the weight of TV term is $w_2 = .4$. Then a range values of the regularization parameter $\lambda \in [0.2, 7.4e9]$ are tested and a L-Curve is formed by the relative path of the parameter value λ. In Figure(5.7), a particular L-shape for the plot of the value of the objective function

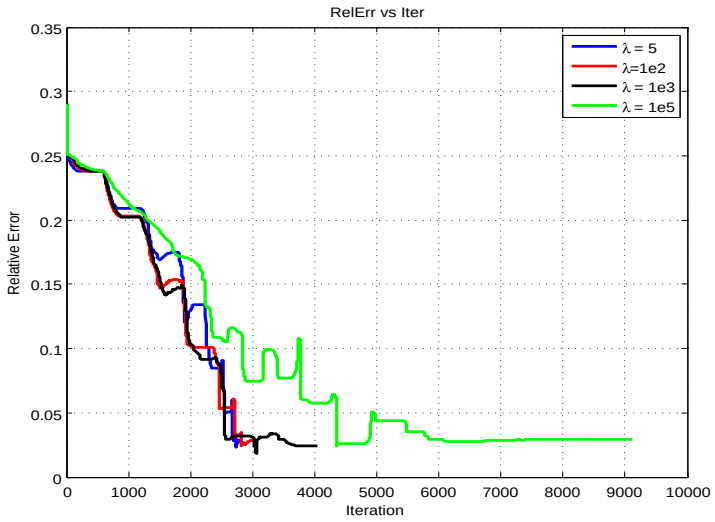

Figure 5.8: The experiments consist of $M = 18$ sensors, $N = 180$ 1° grid points and $SNR = 22dB$. Each case finally detects the true DOA $[30^\circ, 100^\circ, 120^\circ]$ successfully, but the computational load is affected by the regularization parameters, where the bad case costs as three times as the good case.

against the fidelity with respect to various values of $\lambda \in [0.2, 7.4e9]$. In this experiment, 60 trails are taken and around the corner of this plot where the parameter $\lambda = 3.4721$, the tradeoff is balanced optimally. This Figure (5.6) gives an intuitive example about how the parameter value affects the detection result and we notice that in some cases with a bad value of the parameter, the detection fails.

Table 5.1: The convergence speed with various regularization parameter values.

SNR=22dB, $w_1 = 0.6$, $w_2 = 0.4$				
Case	λ	TotaIter	Obj	RelErr(%)
1	5	2855	28.364	1.94
2	1e2	3282	37.300	1.94
3	1e3	4925	43.819	1.79
4	1e5	7900	42.514	1.85

The selection of the regularization parameter can affect the convergence speed. If the parameter is too large, the convergence will be slow due to the amplification of

the fidelity term. In the Figure (5.8) and Table (5.1), we tested four different parameter values: $\lambda = 5, 1e2, 1e3, 1e5$. All these four cases finally cease at the same noise level, but apparently as the parameter value increases, more iterations are required, in other words, the computational load is increased dramatically as λ increases. It is worth to point out that, the case $\lambda = 5$ performs best according to the Table (5.1), and the value $\lambda = 5$ is close to the optimal parameter value estimated from the L-Curve method as Figure (5.7).

5.5.2 Comparison with L1-SVD

The L1-SVD is a method for solving the joint time source localization problem, and it was proposed by D.Malioutov, M.Cretin and A.S.Willsky [MA05] in 2005. It applies the principle component analysis to the joint time measurement and only keeps the signal subspaces by using an user's assumption on the number of the underlying the sources. Then the redundancy introduced by the increased amount of the data in the temporal domain is reduced. This way avoids the increased computational load as more of the samples in the temporal domain are collected. However, on the other hand this scheme requires a prior assumption on the underlying number of sources and this way may not be practical, although the author claims that the algorithm is not so sensitive to this prior guess.

The method L1-SVD contains three main steps: the dimensionality reduction, forming the joint sparsity objective function, and detecting the implied sparsity via ℓ_1-norm minimization. First the $M \times T$ observation matrix Y is decomposed into signal and noise subspaces by the singular value decomposition (SVD). Next, a certain number of dominant singular values are kept, and the remaining less important singular values are dropped, where the amount of the so called important singular values is determined by the user's knowledge of the number of the sources or a simply a guess on it; Then the reduced problem is reformulated into an inverse problem with joint sparsity underlying

variables with much smaller dimensionality than the original one; Finally, the problem is reformulated into a second order cone programming and solve with a SOCP a solver. Suppose $Y \in C^{M\times T}$ denotes a joint time sample $Y = [y(t_1), \cdots, y(t_T)]$ and is decomposed via the SVD $Y = ULV^T$, when there is no additional noise in the sensor, the $\{y(t_i)\}_{i=1}^{T}$ lies in a K-dimensional subspace, where K is the number of sources determined by the user in advance. Then it is reasonable to keep the K subspaces instead of T, where $K \ll T$, to detect the right linear combinations of the column vectors in A, such that the underlying sparse signal can be represented in this way. Mathematically, this process can be expressed as

$$Y_{sv} = ULD_k = YVD_k, \tag{5.37}$$

where $D_k = [\mathbf{I_k} \quad \mathbf{0}]$ is composed by a $K \times K$ identity block and a $K \times (T-K)$ zero block, the similar operation can also be applied to the $M \times T$ underlying matrix S and the noise N, such as, $S_{sv} = SVD_k$ and $N_{sv} = NVD_k$. Now the source localization problem becomes $Y_{sv} = AS_{sv} + N_{sv}$. Since the underlying variable S_{sv} is sparse in the spatial domain (the column), the ℓ_2-norm of the row vectors in S_{sv} is defined as $s_i^{(\ell_2)} = \sum_{j=1}^{K} \sqrt{(S_{ij}^{sv})^2}$, $\forall i$, and the sparsity of the $N \times 1$ vector $s^{(\ell_2)}$ can be estimated via minimizing the l^1-norm regularized problem:

$$\min_{S_{sv}} ||s_i^{(l_2)}||_1 + \lambda ||AS_{sv} - Y_{sv}||^2 \tag{5.38}$$

In [MA05], the transforms posed the problem (5.38) into a second order cone programming and solve with Sedumi. The key steps of the L1-SVD is summarized in Algorithm 10.

Algorithm 10 The $L1-SVD$ procedure

Require: Given the joint time sample $Y = [y(t_1), \cdots, y(t_T)]$

1: Compute the SVD: $Y = ULV'$
2: Reduce the dimensionality: $Y_{sv} \triangleq YVD_k$, $S_{sv} = SVD_k$
3: Forming an ℓ_1-norm regularized inverse problem (5.38)
4: Reformulate (5.38) into second order cone programming and solve it via *SeDuMi*

The robustness of the algorithms

The proposed algorithm shows more stability and robustness in numerical computation than L1-SVD, and in this section we compare the proposed Lp-TV algorithm with L1-SVD in the robustness with respect to the noise level, robustness to the number of snapshots as well as the sensitivity of the two schemes with respect to various values of the regularization parameter λ.

Experiment.1 : We compare the robustness of the two algorithms to the noise. We consider a uniform linear array of $M = 18$ sensors. Three narrowband signals with DOA $[50^\circ, 75^\circ, 132^\circ]$ in the far field impinge on this array, and a total number of snapshots $T = 200$ are taken. The noise level are measured as signal to noise ratio (SNR) defined as

$$SNR_{dB} = 10 * \mathbf{log}_{10}(\frac{||Y||_{fro}}{Var(N)}), \tag{5.39}$$

where $||Y||_{fro}$ is the Frobenius norm of the joint time sample, $Var(N)$ is the variance of the additive noise. This ratio measures the level of desired signal to the level of the background noise. We run both two algorithms to detect the DOA with respect to various SNR and compare the probability of successful detection of the same source locations. Here each of our data points is based on 20 independent trails. In Figure (5.9), the Lp-TV performances more robust with respect to varying noise level, especially in the case of high noise level. We notice that at $SNR = -10dB$, the Lp-TV still can detect the DOA successfully with certain probabilities, but L1-SVD completely fails in the detection when $SNR \leq 2dB$; at $SNR = -5dB$, the Lp-TV can detect the DOA successfully with probability one, however the L1-SVD is still not working at this noise level. Figure (5.9) shows that the proposed algorithm works much better especially under extremely strong background noise, and the range of SNR that the L1-SVD can work with is limited compared with Lp-TV. Although both algorithms work well under mild noise level, we can conclude that the Lp-TV shows more robustness to the noise.

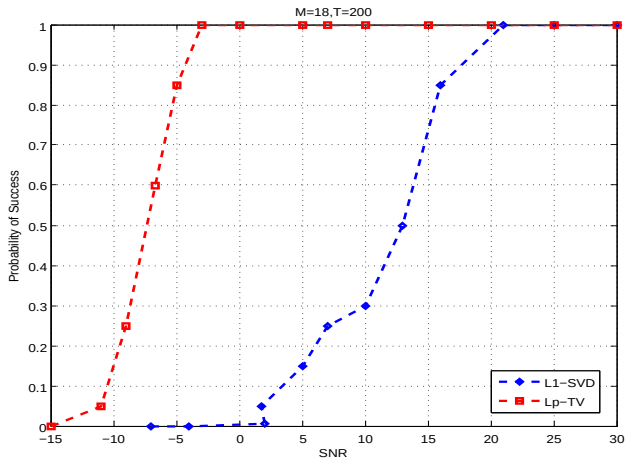

Figure 5.9: The probability of correct detecting three source as a function of SNR, where $M = 18$ sensors, $T = 200$ snapshots and the true DOA $[50°, 75°, 132°]$

Experiment.2 : Next we compare the robustness of the two schemes to the number of snapshots. Intuitively the more snapshots we take, the more accuracy we should have, but this will need more time and the relative cost for collecting the data maybe also increased. In this experiment, we consider both problems for the mild noise level such that the effects of the noise are not taken into account. Let's consider a ULA with $M = 18$ sensors, three narrowband signals from DOA $[50°, 75°, 132°]$ impinge the array, and set $SNR = 21dB$ since from Figure (5.9) both algorithms work well at this level of noise. The experiment is accomplished by doing the detection on the impinging DOA with varying the number of snapshots to compare the probability of success. Each data is based on 20 independent trails. In Figure (5.10), both Lp-TV and L1-SVD show poor ability of detection when a single snapshot is taken. But as more measurements in the temporal domain are involved, at $T = 5$ snapshots, the probability of success in Lp-TV get increases rapidly, while L1-SVD also increases but much slower; The L1-SVD detection can not give a reliable detection until the number of snapshots is over 50, but the Lp-TV scheme can provide a reliable detection results when the number of snapshots reaches $T = 15$.

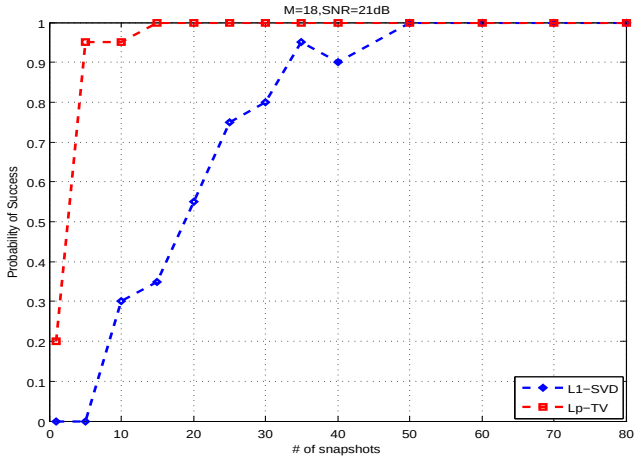

Figure 5.10: The probability of correct detecting three source as a function of number of snapshots, where $M = 18$ sensors, $SNR = 21dB$ snapshots and the true DOA $[50°, 75°, 132°]$

Experiment.3 In this experiment, we test how the regularization parameter λ affects both schemes. In the above discussion, we know that in some cases it is not easy to find an optimal value, although this value plays a key role in the reconstruction. In Figure (5.8), the Lp-TV shows the convergence with respect to a wide range values of $\lambda \in [3, 1e5]$. We set these two schemes in same situation and compare how they react to various λ values. In Table (5.2), each entry is the averaged error based on 20 trials, where the relative error with respect to the regularization parameter λ is defined as

$$RelErr_\lambda = \frac{||\hat{s}_\lambda - s_{true}||}{||s_{true}||}.$$

At the mild noise level($SNR = 40dB$ or $20dB$), for the Lp-TV scheme, the number of snapshots becomes the major factor of affecting the error, such as when $SNR = 40dB$, $T = 200$ snapshots, as the regularization parameter λ varies from 3 to $1e5$, the relative error of Lp-TV varies between $[2.5\%, 3.5\%]$, in other words, this scheme is robust to the regularization parameter; on the other hand, the relative error of L1-SVD varies between $[4.8\%, 8.3\%]$ and it performs more sensitively to the λ value than Lp-TV. When the background noise becomes strong ($SNR = 5dB$), the λ brings more variation

Table 5.2: The comparison of the sensitivity to the regularization parameter λ.

	SNR = 40 dB, M = 18											
	Lp-TV						L1-SVD					
T \ λ	3	5	10	1e2	1e3	1e5	3	5	10	1e2	1e3	1e5
1	0.578	0.432	0.527	0.480	0.312	0.462	0.447	0.4191	0.4265	0.3772	0.4734	0.4221
10	0.121	0.131	0.112	0.129	0.127	0.122	0.119	0.1161	0.1308	0.1744	0.1573	0.1793
100	0.045	0.037	0.037	0.032	0.031	0.053	0.063	0.0494	0.0500	0.0774	0.0920	0.0935
200	0.027	0.025	0.028	0.030	0.029	0.035	0.083	0.0452	0.0487	0.0678	0.0856	0.0827
1000	0.013	0.015	0.013	0.016	0.018	0.021	0.033	0.0332	0.0359	0.0339	0.0477	0.0585
	SNR = 20 dB, M = 18											
	Lp-TV						L1-SVD					
T \ λ	3	5	10	1e2	1e3	1e5	3	5	10	1e2	1e3	1e5
1	0.772	0.593	0.645	0.764	0.927	0.982	0.616	0.652	0.792	0.863	0.734	0.722
10	0.111	0.099	0.121	0.124	0.304	0.316	0.198	0.254	0.332	0.358	0.334	0.339
100	0.025	0.039	0.039	0.039	0.076	0.112	0.072	0.082	0.162	0.191	0.195	0.204
200	0.034	0.029	0.036	0.036	0.062	0.082	0.051	0.071	0.121	0.167	0.161	0.164
1000	0.013	0.015	0.019	0.019	0.040	0.053	0.039	0.056	0.054	0.111	0.120	0.112
	SNR = 5 dB, M = 18											
	Lp-TV						L1-SVD					
T \ λ	3	5	10	1e2	1e3	1e5	3	5	10	1e2	1e3	1e5
1	—	—	—	—	—	—	—	—	—	0.983	—	—
10	0.496	0.668	0.678	0.940	—	—	0.704	0.836	0.922	0.852	0.931	0.885
100	0.032	0.039	0.049	0.299	0.591	0.631	0.446	0.420	0.446	0.431	0.458	0.454
200	0.024	0.028	0.033	0.134	0.404	0.652	0.329	0.365	0.360	0.394	0.372	0.386
1000	0.014	0.014	0.018	0.035	0.146	0.189	0.121	0.189	0.228	0.256	0.261	0.253

to the relative error, such as when $T = 200$ snapshots are taken, the relative error in the Lp-TV scheme varies from 2.4% to 65.2%, and $\lambda = 1e3, 1e5$ are bad options for this case. But, the relative error in L1-SVD maintains at a high level around 32% for all λ values; if we further increase the snapshots to $T = 1000$, both schemes perform well for all λ values, and Lp-TV still has much lower relative error than L1-SVD in this situation. General, according to the Table (5.2), Lp-TV shows more stability and accuracy than L1-SVD with respect to all the λ values we considered in the problem.

5.5.3 The super-resolution in Lp-TV and L1-SVD

In this section we compare the ability of resolving the closely distributed sources with the Lp-TV and L1-SVD. The proposed Lp-TV algorithm shows a good ability to resolve close sources over L1-SVD, due to the use of the Total Variation norm. Let us consider a problem with a ULA consisting of $M = 18$ sensors, the $SNR = 21dB$ and

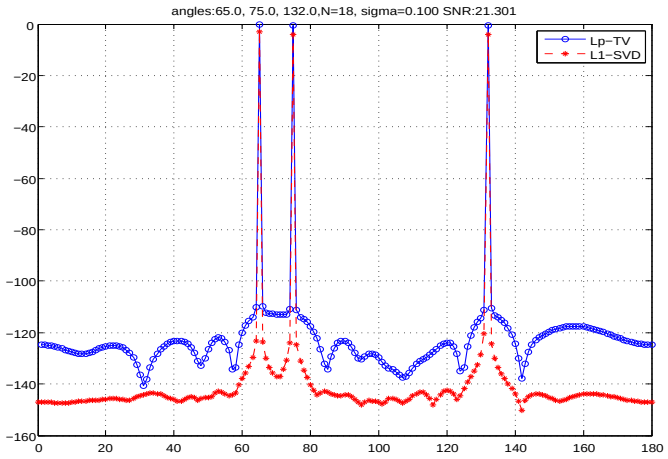

Figure 5.11: Lp-TV and L1-SVD resolve the DOA $[65°, 75°, 132°]$ successfully, where $M = 18$, $SNR = 21dB$, $T = 200$

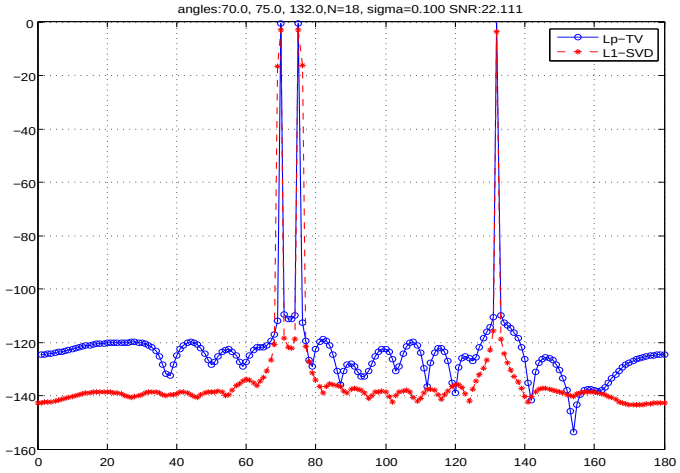

Figure 5.12: Lp-TV and L1-SVD resolve the DOA $[70°, 75°, 132°]$ successfully, where $M = 18$, $SNR = 21dB$, $T = 200$

$T = 200$ snapshots.

First we try to resolve the DOA $[65°, 75°, 132°]$, the two closely spaced sources at $65°$ and $75°$ are $10°$ apart. Both Lp-TV and L1-SVD resolve the sources successfully Figure (5.11). But in Figure (5.12), when we apply the two schemes to detect the DOA $[70°, 75°, 132°]$ where the two closely spaced sources are just $5°$ apart, the Lp-TV

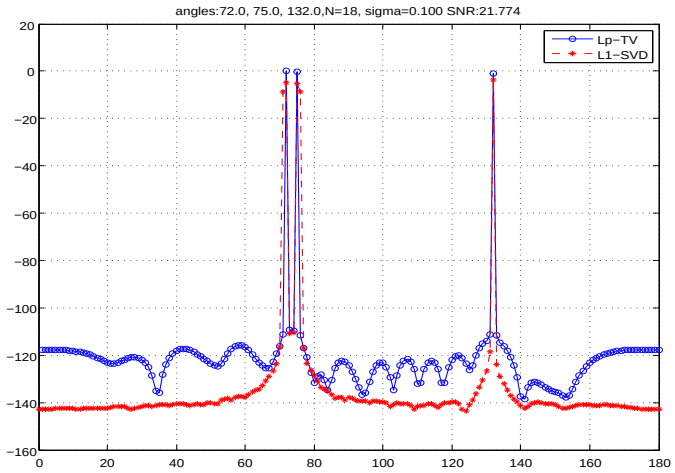

Figure 5.13: Lp-TV and L1-SVD resolve the DOA $[72°, 75°, 132°]$ successfully, where $M = 18$, $SNR = 21dB$, $T = 200$

resolves these $5°$ sources successfully, but a fat lobe appears in the L1-SVD detection and the resolution is apparently not as good as Lp-TV.

Furthermore, we push the DOA of the sources even closer with $3°$ apart and apply the two schemes to resolve the DOA $[72°, 75°, 132°]$. The Figure (5.13) shows the result: the Lp-TV can detect the two close spaced sources and one far source successfully and accurately, but in this case the L1-SVD fails to resolve the two closely spaced sources.

5.6 Conclusion

We present a simple but efficient algorithm using ℓ^p-norm and TV regularization for source localization. We reformulate the ULA detecting problem into an expression of sparsity, and propose a fast and efficient algorithm for detecting the source location. The results of the numerical experiments show that the algorithm is robust to noise and a wide range of regularization parameters. Besides, the comparison with the L1-SVD demonstrates that the proposed algorithm has better properties in the robustness and especially possesses the ability to resolve closely distributed sources.

Chapter 6

Future Work

The optimal regularization parameter selection is still an open problem for nonlinear algorithms and this issue can not be ignored, although we have shown that ALSR is robust to wide range value of r. Another issue is proof of incoherence when applying the CS in sparse MR imaging. Actually, so far the incoherence issue has been proved perfectly only for the random matrices, such as Gaussian random matrix and random Fourier matrix [D.L06] [E.C06], but in MR imaging the random sampling may encounter some difficulties because of the physical constraints and the limits of the equipment; on the other hand, as we know an ideal sampling pattern should take denser samples in the center of the k-space and less dense at the outer parts. The results of the our experiments show that undersampling in k-space will not degrade the quality, but we still need to find a way to show the incoherence theoretically when we use various sampling patterns. The proposed source localization algorithm using TV and ℓ_p-norm results in a high resolution and can identify closely distributed sources perfectly, but due to the use of nonconvex programming we can not guarantee to find a global minimizer.

[AB98] N.Lee A.Chambolle, R.DeVore and B.Lucier. Nonlinear wavelet image processing:variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. Image Process.*, (7):319–1335, 1998.

[AD91] J.W.Kay A.N.Thompson, J.C.Brown and D.M.Titterington. A comparison of methods of choosing the smoothing parameter in image restoration by regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:326–339, 1991.

[A.K91] A.Katsaggelos. *Digital Image Restoration*, volume 23. New York: Springer-Verlage, 1991.

[AM09] A.Beck and M.Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, pages 183–202, 2009.

[A.N63a] A.N.Tikhonov. Regularization of incorrectly posed problems. *Soviet Mathematics Doklady*, 4:1624–1627, 1963.

[A.N63b] A.N.Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Doklady Akademii Nauk SSSR*, 1963.

[APL97] A.Chambolle and P.-L.Lions. Image recovery via total variation minimization and related problems. *Numer. Math.*, 76:167–188, 1997.

[AV97] A.N.Tikhonov and V.Arsenin. *Solutions of Ill-posed Problems*. Wiley, New York, 1997.

[BB88] J. Barzilai and J. Borwein. Two point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(2):141–148, 1988.

[B.K95] B.K.Natarajan. Sparse approximate solutions to linear systems. *SIAM J.Comp.*, 24:227–234, 1995.

[Cha07] Rick Chartrand. Nonconvex regularization for shape preservation. In *IEEE International Conference on Image Processing (ICIP)*, 2007.

[CM96] C.R.Vogel and M.E.Oman. Iterative method for total variation denoising. *SIAM journal on Scientific Computing*, 17(1):227–238, 1996.

[CM98] C.R.Vogel and M.E.Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Transactions on Image Processing*, 7(6):813–824, 1998.

[DA92] J.C.Koch. D.L.Donoho, I.M.Johnstone and A.S.Stern. Maximum entropy and the nearly black object. *J.R.Statist.Soc. B*, 54(1):41–81, 1992.

[dBM07] E.Van den Berg and M.P.Friedlander. Spgl1: A matlab solver for large-scale sparse reconstruction. *Technique Report*, 2007.

[DC97] D.C.Dobson and C.R.Vogel. Convergence of an iterative method for total variation denoising. *Numer. Anal.*, 34:1779–1791, 1997.

[DD93] D.H.Johnson and D.E.Dudgeon. *Array Signal Precessing-Concepts and Techniques*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[D.L62] D.L.Phillips. A technique for the numerical solution of certain integral equations of the first kind. *Journal of the Association for Computing Machinery*, 9:84–97, 1962.

[D.L95] D.L.Donoho. De-noising by soft thresholding. *IEEETransactions on Information Theory*, pages 613–627, 1995.

[D.L06] D.L.Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.

[DM02] D.S.Taubman and M.W.Marcellin. *JPEG 200: Image Compression Fundamentals, Standards and Practice*. Kluwer International Series in Engineering and Computer Science, 2002.

[DM03] D.L.Donoho and M.Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proc. Natl. Acad. Sci. USA*, pages 2197–2202, 2003.

[DM06] D.Model and M.Zibulevsky. Signal reconstruction in sensor arrays using sparse representations. *Signal Processing*, 86(3):624–638, 2006.

[DW05] D.Goldfarb and W.Yin. Second-order cone programming methods for total variation based image restoration. *SIAM J. Sci. Comput.*, 27(2):622–645, 2005.

[DX01] D.L.Donoho and X.Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, pages 2845–2862, 2001.

[DXC06] P.Lin D.Krishnan and X.-C.Tai. An efficient operator splitting method for noise removal in images. *Commun. Comput. Phys.*, 1(5):847–858, 2006.

[EB08] E.Candes and B.Recht. Exact matrix completion via convex optimization. *Convex Optimization. Submitted for publication*, 2008.

[E.B09] M.P.Friedlander E.Berg. Joint-sparse recovery from multiple measurements. *Univ of British Columbia, Technical Report TR-2009-07*, 2009.

[E.C04] T.Tao E.Candes, Romberg. Near-optimal signal recovery from random projections and universal encoding strategies. *IEEE Trans. Inform. Theory*, 2004.

[E.C05] T.Tao E.Candes. Decoding by linear programming. *IEEE Trans.Inform.Theory*, (51):4203–4215, 2005.

[E.C06] T.Tao E.Candes, J.Romberg. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, Feb 2006.

[EJ05] E.Candes and J.Romberg. Practical signal recovery from random projections. *Compressive Sensing Resources, http://www.dsp.ece.rice.edu/cs*, 2005.

[EJ06] E.Candes and J.Romberg. l1 magic: Recovery of sparse signals via convex programming. *http://www.acm.caltech.edu/l1magic*, 2006.

[EY07] W.Yin E.Hale and Y.Zhang. A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. *Technical Report TR07-07, Department of Computational and Applied Mathematics, Rice University, Houston,TX*, 2007.

[G.A94] B.-F.Raud and P.Charbonnier G.Aubert, M.Barlaud. Deterministic edge preserving regularization in computed imaging. *Technical Report 94-01, Informatique Signaux et Systems de Sophia Antipolis, France*, 1994.

[GJ92] G.H.Glover and J.M.Pauly. Projection reconstruction technique for reduction of motion effects in mri. *Magn Reson Med*, 28:275–289, 1992.

[HB77] H.Andrews and B.Hunt. *Digital Image Restoration*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[HJ79] S.Bank H.Taylor and J.McCoy. Deconvolution with the l1 norm. *Geophysics*, 44:49–52, 1979.

[HJ06] M.Nikolova H.Fu, M.K.Ng and J.L.Barlow. Efficient minimization methods of mixed l2 -l1 and l1 -l1 norms for image restoration. *SIAM J. Sci. Comput.*, 27(6):1881–1902, 2006.

[IB97] I.F.Gorodnitsky and B.D.Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, 1997.

[I.D92] I.Daubechies. *Ten lectures on Wavelets*. SIAM, 1992.

[IM04] C.Mol I.Daubechies and M.Defrise. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, (57):14131457, 2004.

[JA91] D.Nishimura J.Jackson, C.Meyer and A.Macovski. Selection of a convolution function for fourier inversion using gridding. *IEEE Trans Med Imaging*, 10(3):473–478, 1991.

[JA05] J.Tropp and A.Gilbert. Signal recovery from partial information via orthogonal matching pursuit. *preprint*, 2005.

[J.A06] J.A.Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Trans. Info. Theory*, 52(3), March 2006.

[J.C69] J.Capon. High resolution frequency wavenumber spectrum analysis. *Proc. IEEE*, 57(8):1408–1418, Aug 1969.

[JDM98] J.Bioucas-Dias and M.Figueiredo. Nonlinear wavelet image processing:variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. Image Process.*, (7):319–1335, 1998.

[JF73] J.Claerbout and F.Muir. Robust modelling of erratic data. *Geophysics*, 38(826-844), 1973.

[J.J96] J.J.Fuchs. Linear programming in spectral estimation. application to array processing. *Proc.IEEE Int.Conf.Acoust.Speech, Signal Process*, 6:3161–3164, 1996.

[J.J01] J.J.Fuchs. On the application of the global matched filter to doa estimation with uniform circular arrays. *IEEE Trans. on signal processing*, 49(4), April 2001.

[J.J04] J.J.Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, June 2004.

[JX06] J.Chen and X.Huo. Theoretical results on sparse representations of multiple-measurement vectors. *IEEE Transactions on Signal Processing*, (53):4634–4643, December 2006.

[JY09] S.Rao J.Wright, A.Ganesh and Y.Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. *submitted to Journal of the ACM*, 2009.

[KJ98] K.Scheffler and J.Hennig. Reduced circular field of view imaging. *Magn Reson Med*, 40(3):474–480, 1998.

[KK99] K.Ito and K.Kunisch. An active set strategy based on the augmented lagrangian formulation for image restoration. *SAIM: Math. Model. Numer. Anal.*, 33:1–21, 1999.

[LE92] S.Osher L.Rudin and E.Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 1992.

[MA05] D.Malioutov M.Cetin and A.S.Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE Transactions on Signal Processing*, 53(8):3010–3022, 2005.

[MD79] M.R.Garey and D.S.Johnson, editors. *Computers and Intractability: A guide to the theory of NP-completeness*. New York: W.H.Freeman, 1979.

[M.E06] M.Elad. Why simple shrinkage is still relevant for redundant representations? *IEEE Trans. Inform. Theory*, 52(12), December 2006.

[MJ07a] D.Donoho M.Lustig and J.M.Paul. Sparse mri:the application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine*, 2007.

[MJ07b] J.Santos M.Lustig, D.Donoho and J.Pauly. Compressed sensing mri. *IEEE Signal Processing Magazine*, 2007.

[MM07] B.Matalon M.Elad and M.Zibulevsky. Subspace optimization methods for linear least squares with non-quadratic regularization. *Appl. Comput. Harmon. Anal.*, (23):346–367, 2007.

[MP93] M.Hanke and P.C.Hansen. Regularization methods for large scale problems. *Surveys on Mathematics for industry*, pages 253–315, 1993.

[MP98] M.Bertero and P.Boccacci. *Introduction to Inverse Problems in Imaging*. Bristol,UK.:IOP, 1998.

[MR03] M.Figueiredo and R.Nowak. An em algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.*, 12:906–916, 2003.

[MS07a] R.Nowak M.Figueiredo and S.Wright. Gradient projection for sparse reconstruct io: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4), 2007.

[MS07b] R.Nowak M.Figueiredo and S.Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal on Selected Topics in Signal Processing*, 1, 2007.

[MXC04] O.S. M.Lysaker and X.-C.Tai. Noise removal using smoothed normals and surface fitting. *IEEE Trans. Image Process.*, 13(10):1345–1357, 2004.

[MY08] M.Mishali and Y.C.Eldar. Reduce and boost: Recovering arbitrary sets of jointly sparse vectors. *IEEE Transactions on Signal Processing*, 56(10), October 2008.

[N.K84] N.Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[N.M89] N.Megiddo. *Pathways to the optimal set in linear programming. In Progress in Mathematical Programming: Interior-Point and Related Methods*. 1989.

[P.C92] P.C.Hansen. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Rev.*, 34:561–580, 1992.

[P.C98] P.C.Hansen. *Numerical Aspects of Linear Inversion*. SIAM Philadelphia, 1998.

[PD93] P.C.Hansen and D.P.O'Leary. The use of the l-curve in the regularization of discrete ill-posed problems. *SIAM Journal on Scientific Computing*, 14:1487–1503, 1993.

[RI84] R.Monteiro and I.Adler. Interior path following primal-dual algorithms.part i: Linear programming. *Mathematical Programming*, 44:27–41, 1984.

[RM01] R.Nowak and M.Figueiredo. Fst wavelet-based image deconvolution using the em algorithm. *Proceedings of the 35th Asilomar conference on Signals*, 2001.

[R.O81] R.O.Schmidt. *A signal subspace approach to multiple emitter location and spectral estimation*. PhD thesis, Stanford Univ., 1981.

[Roc70] R.Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[R.V02] Curtis R.Vogel. *Computational methods for inverse problems*. Society for industrial and applied mathematics, 2002.

[SD07] M.Lustig-S.Boyd S.Kim, K.Koh and D.Gorinvesky. A method for large-scale l1 regularized least squares problems with applications in signal processing and statistics. *Tech.Report,Dept.of Electrical Engineering, Stanford University*, 2007.

[SKD05] K.Engang S.F.Cotter, B.D.Rao and K.Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, (53):2477–2488, July 2005.

[S.M99] S.Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[SM08] R.Nowak S.Wright and M.Figueiredo. Sparse reconstruction by separable approximation. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

[S.N01] Parbhakar S.Naidu. *Sensor array signal processing*. CRC Press, 2001.

[SP81] S.Levy and P.Fullagar. Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution. *Geophysics*, 46(1235-1243), 1981.

[S.S99] M.A.Saunders S.S.Chen, D.L.Donoho. Atomic decomposition by basis pursuit. *SIAM J.Scientific Computing*, 20:31–61, 1999.

[SW86] F.Symes Santosa and W.W. Linear inversion of band limited reflection seismograms. *SIAM J.Sci.Statist.Comput.*, 7:1307–1330, 1986.

[T.F88] T.F.Chan. An optimal circulant preconditioner for toeplitz systems. *SIAM J.Sci.Stat.Comput*, 9:766–771, 1988.

[TK06] T.Chan and K.Chen. An optimization based total variation image denoising. *Multi-scale Model. Simul.*, 5(2), 2006.

[TP99] G.H.Golub T.F.Chan and P.Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J.Sci.Stat.Comput*, 20(6), 1999.

[VG08] M.F.Duarte V.Cevher and G.Baraniuk. Distributed target localization via spatial sparsity. In *16th European Signal Processing Conference in 2008*. 16th European Signal Processing Conference in 2008, 2008.

[VR08] J.H.McClellan V.Cevher, A.C.Gurbuz and R.Chellappa. Compressive wireless arrays for bearing estimation. *IEEE Int. Conf. on Acoustics*, 2008.

[WY08] Yin Zhang Wotao Yin. Extracting salient features from less data via l1 minimization. *SIAG/OPT Views-and-News*, 10(1):11–19, March 2008.

[Y.E07] Y.E.Nesterov. Gradient methods for minimizing composite objective function. *CORE report available at http://www.ecore.be/DPs/dp 1191313936.pdf*, 2007.

[YF96] Y.Li and F.Santosa. A computational algorithm for minimizing total variation in image reconstruction. *IEEE Trans. Image Process.*, 5:987–995, 1996.

[YY08] W.Yin Y.Wang, J.Yang and Y.Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imaging Sci*, pages 248–272, 2008.

Appendix A

THE SOURCE CODE OF ALSR

A.1

```
%The demo for the sparse reconstruction
% demo_ALSR.m
clc;clear;
fprintf('Preparing for the initial data')
%read the phantom for simulation
Im = phantom(128);
Im = Im./max(Im(:));
[m,n] = size(Im);
%Generate the sampling operator
index = fftshift(MRImask(m,30));
index=find(index);
Mea = length(index);
Selec=speye(m*n);Selec=Selec(index,:);
fprintf('\n %i%% fourier data are used'...
,round(100*Mea/m/n))
%generate the random sample in k space
FIm=fft(Im(:))/sqrt(m*n);
f = Selec*FIm;
%generate the white noise
sigma =0.001;
noise=sigma*(randn(Mea,1)+sqrt(-1)...
*randn(Mea,1));
%random partial fourier data
f = f + noise;
%initial guess
u0 = real(ifft(Selec'*f)*sqrt(m*n));
% Wavelet transformation operator
W = @(x) Wavedb1Phi(x,1);
WT = @(x) Wavedb1Phi(x,0);
%-----------------------------------------
%Set up all the parameters
aTV = 2;aL1 = 1;r=1e2;
larg = ones(length(f),1);
%parameter for updating the larg
rho = 4*r;
%the solver
[u_cg,ImError_cg,Total_Iter_cg,Obj_cg,Fid_cg,Im_t_cg]...
= solver_cg(m,n,r,Selec,W,WT,u0,f,aTV,aL1,larg,rho);
[u,ImError,Total_Iter,Obj,Fid,Im_t] ...
= solver_Precon(m,n,r,Selec,W,WT,u0,f,aTV,aL1,larg,rho);
fprintf('\n');
U_cg=reshape(u_cg,[m,n]);U=reshape(u,[m,n]);
U0=reshape(u0,[m,n]);
SNR_init = snr(U0,Im);
```

```
SNR_final = snr(U,Im);
fprintf('Initial SNR:%f dB',SNR_init);
fprintf('\n Enhanced SNR:%f dB\n',SNR_final);
figure(1)
plot(Im_t_cg,ImError_cg/norm(Im),'--','LineWidth',2)
hold on
plot(Im_t,ImError/norm(Im),'LineWidth',2)
title('Reconstruction Error')
xlabel('CPU Time')
ylabel('MSE')
grid on
legend('CG Iter','CG-Precon Iter')
set(gca,'FontName','Times','FontSize',16)
hold off
figure(2);
subplot(221); imshow(Im,[]);
title('Original');
subplot(222); imshow(U0,[]);
title(sprintf('Back Projection SNR:%ddB'...
    ,round(SNR_init)));
subplot(223); imshow(U_cg,[]);
title(sprintf('CG-Reconstructed SNR:%ddB'...
    ,round(SNR_final)));
subplot(224); imshow(U,[]);
title(sprintf('CG-Precon SNR:%ddB'...
    ,round(SNR_final)));
figure(3)
plot(Im_t_cg,Fid_cg,'--','LineWidth',2)
hold on
plot(Im_t,Fid,'LineWidth',2)
title('Reconstruction Error')
xlabel('CPU Time')
ylabel('Fid')
grid on
legend('CG Iter','CG-Precon Iter')
set(gca,'FontName','Times','FontSize',16)
hold off
```

A.2

```
%genData.m
%Generating the TV operator and the
%BCCB preconditioner
function [D1,D2,diagBTTB,T_fcolrow,BTTB_fcolrow]...
=genData(m,n,Selec)
% Input
```

```
% m,n:the size of the imamge
% Selec: the partial fourier operator
% Output
% D1 D2: the row and column difference
%T_fcolrow:the first col and row of each
%block in BTTB
% diagBTTB: the diagnal entries of BTTB
% BTTB_fcolrow: the approximated frist
 col and row of BTTB
% for generating the preconditioner
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 fprintf('\n Preparing for the Data ......\n');
% generate the TV operator
e=ones(n^2,1);
%-----------------------------------------------
%D1 D2 are the column and row difference
%operator (good for large scale probem) with
%peridoic boundary condition
%-----------------------------------------------
%column difference operoter
D1=spdiags([-e,e,e],[0,n,-(n^2-n)],n^2,n^2);
%row difference
D2=spdiags([-e,e,e],[0,1,-(n-1)],n,n);
Mtemp=D2;MMtemp=D2;
for i=1:n-1
    %concatenate to block diagnal
    Mtemp=blkdiag(blkdiag(Mtemp),MMtemp);
end
D2=Mtemp;
%BTTB matrix
DiffOper = D1'*D1+D2'*D2;
diagDiff = diag(DiffOper);
diagBTTB = spdiags(diagDiff,0,m^2,n^2);
DiffOper = DiffOper - diagBTTB;
MatEig = DiffOper + speye(m*n);
t = zeros(2*m,2*n);
for i = 1:n
    t(1:m,i) = MatEig((i-1)*m+1:i*m,1);
    t(2*m:-1:m+2,i) = MatEig((i-1)*m+1,2:m);
end
for i = 1: n-1
    t(1:m,(n+1)+i) = MatEig(1:m,m*(n-i)+1);
    t(2*m:-1:m+2,(n+1)+i)...
= MatEig(1,m*(n-i)+2:m*(n-i+1));
```

```
end
AtA = Selec'*Selec;
% the first colum and first row of F'AF
ifft_col = sum(AtA,1);
ifft_col = full(ifft_col);
fcol = ifft(ifft_col);%*(m*n);
fcol = fcol(:);
frow = conj(fcol); %frow(1)= 0;
% the first colum and first row of F'AF
fcolrow = real([fcol,frow]);
w = zeros(2*m,2*n);
for i = 1:n
    w(1:m,i) = fcolrow((i-1)*m+1:i*m,1);
    if (i == 1)
        w(2*m:-1:m+2,i) = fcolrow(2:m,2);
    else
        w(2*m:-1:m+2,i)...
= fcolrow((i-1)*m:-1:(i-2)*m+2,1);
    end
end
for i = 1:n-1
    w(1:m,2*n-i+1) =...
        fcolrow(i*m+1:-1:(i-1)*m+2,2);
    w(2*m:-1:m+2,2*n-i+1) =...
        fcolrow(i*m+2:(i+1)*m,2);
end
T_fcolrow = real(t + w);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%BTTB: used for generating the preconditioner
BTTB = DiffOper+speye(m*n);
BTTB = BTTB + diagBTTB;
t = zeros(2*m,2*n);
for i = 1:n
    t(1:m,i)=BTTB((i-1)*m+1:i*m,1);
    t(2*m:-1:m+2,i)=BTTB((i-1)*m+1,2:m);
end
for i = 1: n-1
    t(1:m,(n+1)+i) = BTTB(1:m,m*(n-i)+1);
    t(2*m:-1:m+2,(n+1)+i)...
= BTTB(1,m*(n-i)+2:m*(n-i+1));
end
%the first column and first row
%from each toeplitz block in BTTB
BTTB_fcolrow = real(t + w);
```

A.3

```
 function y = MRImask(n,beams)
% produces the fan MRI mask, of size n*n,
% beams is the number of angles
m = ceil(sqrt(2)*n);
aux = zeros(m,m); ima = aux;
aux(round(m/2+1),:) = 1;
angle = 180/beams;
angles = [0:angle:180-angle];
for a = 1:length(angles)
    ang = angles(a);
    a = imrotate(aux,ang,'crop');
    ima = ima + a;
end
ima = ima(round(m/2+1)...
- n/2:round(m/2+1) + n/2-1,...
round(m/2+1) - n/2:round(m/2+1) + n/2-1);
y = (ima > 0);
```

A.4

```
 % Solver
function [u,ImError, Total_Iter,Obj,Fid,Im_t] ...
    = solver_Precon(m,n,r,Selec,...
    W,WT,u0,f,aTV,aL1,larg,rho)
% u0: the initial image
% W,WT: are the wavelet transform operator
% m,n: are the size of the imamge
% beta,h: the penalty parameters
% Selec: the selection matrix
% f: the random sample in the k space
% larg: the largrage multiplier
% rho: the parameter for updating
%u = u0;
N = length(u0);
[D1,D2,diagBTTB,T_fcolrow,BTTB_fcolrow]...
   =genData(m,n,Selec);
%-------------------------------------------%
Total_Iter = 0;
Stop_Cri = 1e-2;
Inner_Tol = 0;
InnerIter_max = 8000;
%-------------------------------------------%
fprintf('\n Now the data is ready .....');
fprintf('\n ---------------------------------');
fprintf('\n----------It start !----------');
```

```
fprintf('\n------------------------------------');
time = cputime;
Inniter = 0; w01=0;w02=0;
while (Inniter < InnerIter_max) && (~Inner_Tol)
w1 = sign(D1*u0(:)).*max(abs(D1*u0(:)) -aTV/r,0);
w2 = sign(D2*u0(:)).*max(abs(D2*u0(:)) -aTV/r,0);
A = reshape(u0,[m,n]);
PsiTU = WT(A);
PsiTU = PsiTU';
v = sign(PsiTU).*max(abs(PsiTU)-aL1/r,0);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the conjugate gradient solver(all 3 solvers work!!)
%u1 = cg(r,w1,w2,v,Selec,...
%     D1,D2,u0,f,W,aTV,aL1,r,larg);
u1 = cgBTTB(r,w1,w2,v,Selec,...
     D1,D2,u0,f,W,larg,diagBTTB,T_fcolrow);
%update the largange multiplier
larg = larg + rho*(Selec*fft(u1(:))./sqrt(N)-f);
Inniter = Inniter+1;
ImError(Total_Iter+Inniter) = Im_Error(u1,n);
Im_t(Total_Iter+Inniter) = cputime-time;
[Obj(Total_Iter+Inniter),Fid(Total_Iter+Inniter)]...
     =ObjEval(D1,D2,u1,Selec,WT,f,aTV,aL1);
fprintf('\n In %ith iteration the Objective function %f\
,the error is %f',...
     Total_Iter+Inniter,Obj(Total_Iter+Inniter)...
     ,ImError(Total_Iter+Inniter));
%check the stopping criteria
tau = stop_tol(w1,w2,w01,w02,u0,u1);
Inner_Tol = (tau <=Stop_Cri);
w01 = w1; w02 = w2;u0 = u1;
end % end inner iteration
Total_Iter = Total_Iter + Inniter;
u=u0;
%end %end outter iter
time = cputime-time;
fprintf('\n------------------------------------');
fprintf('\n-------------It is done!-----------');
fprintf('\n------------------------------------');
fprintf('\n The elapsed time is %f', time);
fprintf('\n The total number of...
iterations:%i',Total_Iter);
fprintf('\n The objective: %f', Obj(Total_Iter));
fprintf('\n The fidelity: %f', Fid(Total_Iter));
fprintf('\n The r is: %e',r);
```

A.5

```
%the preconditioned conjugate gradient solver
function u = cgBTTB(lambda,w1,w2,v,Selec,...
    D1,D2,u0,f,W,larg,diagBTTB,T_colrow)
tev = gentev(T_colrow);
maxIter = 20000;
tol = 1e-1;
u = u0;N = length(u);
PsiV=W(v);PsiV=PsiV(:);
b=D1'*w1+D2'*w2;b=b+PsiV;
b=b+real(ifft((Selec'*f*sqrt(N))));
A_trans_p =...
real(ifft((Selec'*larg*sqrt(length(f)))));
b = b-(1/lambda)*A_trans_p;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = b-tx_2(tev,diagBTTB,u);
e(1) = norm(r);
iter = 1;
t1 = 1;
d = zeros(N,1);
while (iter<maxIter) && (e(iter)/e(1)>tol)
    z = r;
    t1old = t1;
    t1 = z'*r;
    beta = t1/t1old;
    d = z + beta*d;
    s = tx_2(tev,diagBTTB,d);
    suma = d'*s;
    tau = t1/suma;
    u = u + tau*d;
    r = r - tau*s;
    iter = iter + 1;
    e(iter) = norm(r);
end
if (iter == maxIter)
    fprintf('\n Max iterations reached');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function tev = gentev(T_colrow)
tev = fft2(T_colrow);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function y = tx_2(tev,diagBTTB,vec)
%The fast matrix and vector product
y1 = diagBTTB * vec;
```

```
[n1,m1] = size(tev);
m = m1/2;
n = n1/2;
v = reshape(vec,n,m);
ev = zeros(n1,m1);
ev(1:n,1:m) = v;
y = fft2(ev);
y = tev.*y;
y = ifft2(y);
y = y(1:n,1:m);
y = reshape(y,m*n,1);
y = real(y);
y = real(y+y1);
```

Appendix B

THE SOURCE CODE OF LPTV

B1

```
% demo_source.m
clear;clc;
M = 18;%number of sensors
T = 200;%number of time samples
[A,Y,theta_s,sourceAngle,snr]...
= modelGen(M,T);
%dimension reduction
y_av = sum(Y,2)/T;%y_av is size NX1
%set the regularization parameters
alpha=.6;
beta = .4;
lambda = 2;
tic;
e_lp = cgsolver_source(A,y_av,alpha,beta,lambda,.1);
toc;
m_e_lp = 1/max(abs(e_lp));
e_l_p = e_lp.^2*m_e_lp^2;
angles = theta_s*180/pi;
DOA = 10*log10(abs(e_l_p)');
% plot the DOA
plot(angles, DOA, 'bx-');
title(strcat('angles: ',...
sprintf('%.1f, ',180*[sourceAngle]/pi), ...
sprintf('N=%d,SNR:%.3f',M,snr)));
legend('Lp-reg');
figure(gcf);
```

B2

```
%generating the array output
function [A,Y,theta_s,sourceAngle,snr]...
=modelGen(M,T)
%Input:
%M:the number of sensors in the array
%T:number samples in temproal domain
%Output:
% A: the sensing matrix
% Y: the multiple measurement sample
% s_true: the true DOA (reference)
% D1: the column difference operator
w = 2*pi*250;
c = 330;%Speed of the waves in the medium
% generate the signal
sourceW(1) = w;
sourceAngle(1) = 35*pi/180;
```

```
sourceW(2) = w;
sourceAngle(2) = 100*pi/180;
sourceW(3) = w;
sourceAngle(3) = 120*pi/180;
fprintf('True angles: \n');
disp([sourceAngle]*180/pi);
for sig = 1:length(sourceW)
sourceK_z(sig) = -sourceW(sig)*...
cos(sourceAngle(sig))/c;
end
%number of sensors
arrayN=M;
% sensor distance to avoid aliasing use max w
arrayD=2*pi*c/(2*max([sourceW]));
%sensor positions
arrayP_z=((0:arrayN-1)-(arrayN-1)/2)*arrayD;
% sensor noise
arraySigma =.1;
t_samples = T;
disp('Zero Mean');
%for time-average versions
x = rand(length(sourceW),t_samples);
% incoherence case
K = eye(length(sourceW));
x=K*x;
for sig = 1:length(sourceW)
    weight(:,sig) = exp(j*(arrayP_z')*...
        sourceK_z(sig)); %*(1/array.N);
end
arrayY = (weight*x)';
noise = arraySigma/sqrt(2)*...
    (randn(size(arrayY))+j*randn(size(arrayY)));
snr = -10*log10(arraySigma^2/ (norm([real(arrayY)...
    imag(arrayY)], 'fro')^2/(2*prod(size(arrayY)))));
arrayY_clean = arrayY; %%% for debugging
arrayY = arrayY_clean + noise;

%generate the sampling grid
dtheta=1;
theta_s=(0:dtheta:180)*pi/180;
N = length(theta_s); % the number of sampling grids
k_z_s = -max(sourceW)*cos(theta_s)/c;
A=exp(j*(arrayP_z')*k_z_s); % the M by N sensing matrix
Y = arrayY';
```

B3

```
% cgsolver_source.m
function rec=...
cgsolver_source(A,y,alpha,beta,lambda,p)
%-------------------------------------------------
% min_s  alpha ||s||^p_p + beta* TV(s) +
%\frac{lambda}{2} ||As-y||^2_2
%-------------------------------------------------
% A:the sensing matrix
% y:the sample
% alpha:parameter of the lp-norm
% beta:parameter of the finite difference term
% lambda:the parameter of the fidelity term
% p: the l_p norm


mu = 1e-7; %smooth parameter
s = A'*y;
%generating the finite difference operator
n = length(s);
e=ones(n,1);
Dcol = spdiags([-e,e],[-1,0],n,n);
 %Dcol: the finite difference operator
Dcol(1,n)= -1;
Tol = 1e-3;
stpc = 1;
outerIter = 0;

while  stpc
%update the iteration matrix
CgMat = lambda*(A'*A);
RHS = lambda*(A'*y);
CgMat = CgMat + beta*GradTV(s,Dcol,mu);
w = GradLp(s,p,mu);
LHS = CgMat + alpha*w;
objFun = objEval(A,Dcol,s,y,alpha,beta,lambda,p);
fprintf('at the %dth round outer iteration
 the objective is %e\n', outerIter+1,objFun);
rec_s = semi_cgIter(LHS,RHS,s);
if norm(rec_s - s,2) < Tol
    stpc = 0;
end
s = rec_s;
outerIter = outerIter + 1;
end
```

```
rec = s;

function grad_TV = GradTV(s,Dcol,mu)
D = Dcol;
%grad_TV = 2*(D'*D);
diagPsi = sqrt((D*s).^2+mu);
diagPsi = diag(diagPsi);
grad_TV = D'*diagPsi*D;
%grad_TV = (D'*D)/sqrt(s'*(D'*D)*s);

function gradL_p = GradLp(s,p,mu)
% generating the iteration matrix for lp norm
diag_i = p./((abs(s)).^2 + mu).^(1-p/2);
size_diag = length(diag_i);
gradL_p = spdiags(diag_i,0,size_diag,size_diag);

function obj = objEval(A,Dcol,s,y,alpha,beta,lambda,p)
obj = sum((abs(s)).^p);
obj = alpha*obj + beta*(norm(Dcol*s,2))^2;
obj = obj + lambda*norm(A*s-y,2)^2;


function res = semi_cgIter(LHS,RHS,s)
r0 = LHS*s - RHS;
p0 = -r0;
InnerIter = 0;
maxIter = 600;
stopic = 1;
stopTol = 1e-5;
while stopic
    a = (r0'*r0)/(p0'*LHS*p0);
    s1 = s + a*p0;
    r1 = r0 + a*LHS*p0;
    beta = (r1'*r1)/(r0'*r0);
    p1 = -r1 + beta*p0;
    InnerIter = InnerIter + 1;
    normR1 = norm(r1);
    if (normR1 <= stopTol)
        stopic = 0;
   fprintf('The iteration converges
at %dth steps with relative err %e\n',InnerIter,normR1);
    end
    if InnerIter >= maxIter
        stopic = 0;
        fprintf('The max iteration
```

```
is reached, not converge\n');
    end
    p0 = p1;
    r0 = r1;
    s = s1;
end
res = s;
```

BIOGRAPHICAL SKETCH

Wei Shen was born in Beijing, China in 1982. After graduating from the high attached to Tsinghua University, Beijing, he started his undergraduate education in 2001 at Hunan University, Changsha, China. He received his bachelor in Computational Sciences and Mathematics in 2005. He accepted the offer from Arizona State University to study for the PhD in Applied Mathematics with a full tuition waiver and scholarship from the School of Mathematical and Statistical Sciences. In May 2008 Wei Shen received the Master of Arts in Mathematics. From August to December 2010 he was invited to take part in a special program in optimization by the Institute for Pure and Applied Mathematics at the University of California at Los Angeles. He finished his PhD in May 2011. Wei Shen is a student member of the American Mathematical Society and the Society of Industrial and Applied Mathematics.