SPARSE SEISMIC SIGNAL PROCESSING USING ADAPTIVE DICTIONARIES

A Thesis
Presented to
The Academic Faculty

by

Lingchen Zhu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2016

SPARSE SEISMIC SIGNAL PROCESSING USING ADAPTIVE DICTIONARIES

Approved by:

Professor James H. McClellan,
Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Ghassan AlRegib
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Zhigang Peng
School of Earth and Atmospheric Sciences
Georgia Institute of Technology

Professor Justin K. Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Waymond R. Scott
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: August 5, 2016

This dissertation is dedicated my beloved wife,

Shirley Xin,

and my parents,

Yimin Zhu and Yuhua Chen.

Thank you for your love and support.

ACKNOWLEDGEMENTS

During my long journey in the Georgia Institute of Technology, I have received support and encouragement from a great number of people. Without them I can hardly achieve this monumental goal. At this time, I would like to take this opportunity to express my gratitude and appreciation to all of them.

First, I would like to express my special appreciation to my advisor, Prof. James H. McClellan. I would like to thank him for encouraging my research, keeping me excited on the projects and giving me sufficient freedom to grow as an independent researcher. He has been a tremendous advisor, not only does he pass abundant knowledge and expertise but he also sets an exemplar of what a real scholar is. His decade-long career in research and education is a well-prepared armory for students, and I can always expect a handy weapon when I need to attack a problem. In particular, his sharp scientific insights for multidisciplinary research and his ability to elegantly explain complicated ideas always amaze me. Without his guidance and mentorship, I could end up taking many detours in my study.

Then, I would like to thank Prof. Guotong Zhou for her great efforts on running the GT-Shanghai Dual-Master Program. She provided me the opportunity to be enrolled by this program when I was still pursuing my Master's degree in Shanghai Jiao Tong University. This program prepared me well for pursuing a Ph.D. degree later. When I was graduating from this program, she also strongly recommended me to join Prof. McClellan's group. Without her efforts, I would probably never have a chance to be here.

I would like to thank Prof. Ghassan AlRegib, Prof. Zhigang Peng, Prof. Justin Romberg and Prof. Waymond Scott, Jr., who spend their precious times in serving

my dissertation committee and provide me valuable feedbacks on my existing works. Meanwhile, I would like to thank the postdoctoral fellow in the research group, Dr. Entao Liu, who introduced me to the world of seismology. With his solid mathematical background, he answered my questions and eliminated my confusions when I was learning the basic geophysical concepts and working on their computer implementations. Also, I would like to thank all other current members in the group: Lijun Zhu, Anupama Govinda Raj, Peter Tuuk and Gregory Krudysz, and those members who have already graduated: Chenchi Luo, Kyle Krueger, Carson Wick, and Mu-Hsin Wei. Their valuable discussions and suggestions helped a lot for my study and I would never forget the time working with them.

I would like to thank The Center for Signal and Information Processing (CSIP), TI Leadership University Program and The Center for Energy and Geo Processing (CeGP) for sponsoring my research in Georgia Institute of Technology. Also, I would like to thank InterDigital, Inc. and Schlumberger-Doll Research Center for providing me research internship opportunities. Those two fruitful summers inspired me a lot and led to multiple publications.

Finally, but the most importantly, I would like to express my sincerest gratitude to my wife, Shirley. She has been with me since the year of 2001 and we have seen each other growing from high school students all the way up to professionals. When I think troubles come, she always gives me support and helps me make right decisions. No matter how things become, her consistent love raises me up to conquer any challenges we would meet. I must also thank my parents and parents-in-law for their unconditional love and support. Their warm encouragement and heartfelt blessings keep driving me work hard towards every goal in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Seismic surveys have become the primary measurement tool of exploration geophysics, both onshore and offshore, with significant signal processing needed to estimate the properties of earth subsurface via seismic wave propagation. The typical workflow for seismic includes three phases: acquisition, imaging, and interpretation. A high-quality imaging result for interpretation necessitates accurate data acquisition and efficient imaging algorithms. However, seismic data gathers may suffer from noisy and missing traces during acquisition which could possibly limit their use in the following imaging phase. As a convincing quantitative imaging technique, full waveform inversion (FWI) searches for the correct velocity model that can match the acquired seismic dataset. However, due to the high dimensionality of the model space, FWI is inherently a challenging problem, so that regularization techniques are typically applied to yield better posed models. Moreover, FWI also suffers from its prohibitive computational costs that mainly arise from forward modeling of the seismic wavefield for multiple sources at each iteration of a nonlinear minimization process. The dimensionality of the problem and the heterogeneity of the medium both stress the need for faster algorithms and sparse regularization techniques to accelerate and improve imaging results.

This thesis presents a new reconstruction method to mitigate noise and interpolate missing traces in the acquired seismic dataset, as well as a new FWI framework to estimate subsurface models more accurately and efficiently. Both contributions involve sparse approximation of various types of data with respect to adaptive dictionaries that are learned by different strategies. The new seismic data reconstruction method involves a sparse representation over a parametric dictionary, which bridges a gap between model-based and data-driven sparse approximations. The new FWI framework adapts velocity model perturbations to orthonormal dictionaries that are trained in an online manner, and then exploits compressive sensing to significantly reduce the computational cost by requiring many fewer calculations of the forward model. Numerical experiments on synthetic seismic data and velocity models indicate that the new methods can achieve better performance compared to other state-of-the-art methods.

CHAPTER I

INTRODUCTION

1.1 High-resolution Earth Model Imaging

The earth is a complex and heterogeneous medium with properties ranging from the mineral composition scale ($\simeq 10^{-6}\,\mathrm{m}$) to the global scale ($\simeq 10^{6}\,\mathrm{m}$). Exploration geophysics is the study of the earth model through physical methods, such as seismic, gravitational, magnetic, electrical and electromagnetic, using sensors at or near the surface of the earth to elucidate and detect the underlying structures of its subsurface. These methods play a critical role in the oil and gas industry as they are frequently used to identify reservoir characteristics such as faults and traps. Drilling for oil is expensive gambling. With project costs increasing year by year, an oil company could lose a large investment when exploring or developing a field that fails to yield hydrocarbons at profitable rates. To hedge these risks, sophisticated measurements are used to estimate the potential profitability of a field as early as possible in the development process.

1.1.1 Seismic Methods and Related Systems

Seismic methods are widely used to explore the earth's subsurface, in order to give oil companies a more astute indication about the production potential of a field. A seismic survey is always conducted to build a better picture of the hydrocarbon content in a reservoir before actual drilling commences. There are four stages in a seismic survey: (1) seismic acquisition, (2) seismic data preprocessing, (3) seismic migration, and (4) image interpretation. Figure 1.1 depicts an overall field setup for a seismic survey conducted on land. A seismic source, such as vibroseis (attached to a truck for land surveys), or an airgun (attached to a vessel for marine surveys), is used to generate seismic waves. Seismic waves are transmitted from the source and reflect from different rock layers when they travel between layers where rock properties change. Seismic receivers such as geophones (for land surveys) and hydrophones (for marine surveys) are deployed on the surface to record seismic waves in the form of data traces, which contain different wave fronts corresponding to various interactions of the background wavefield with heterogeneities in the earth's subsurface. The resulting seismic dataset, after carefully preprocessing, is then used for seismic migration to obtain a reliable image of the subsurface that describes the properties of deep underground geological structures.

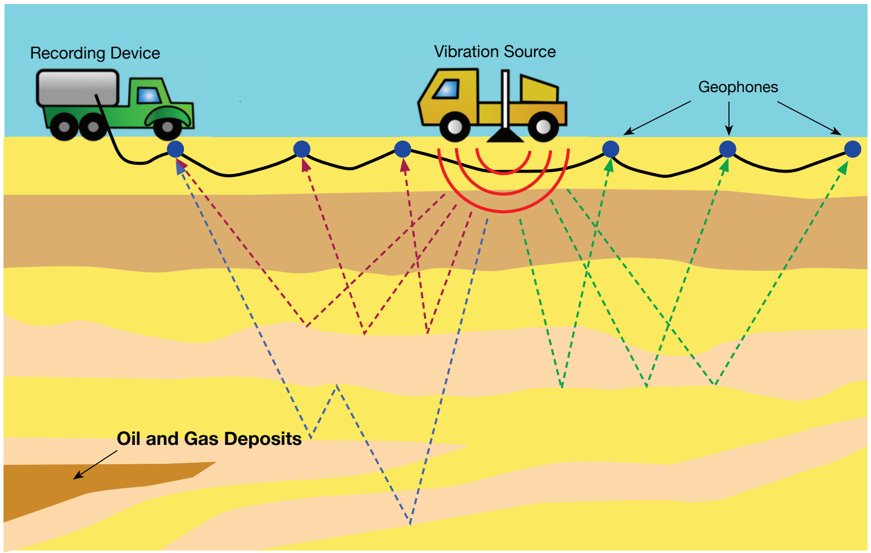
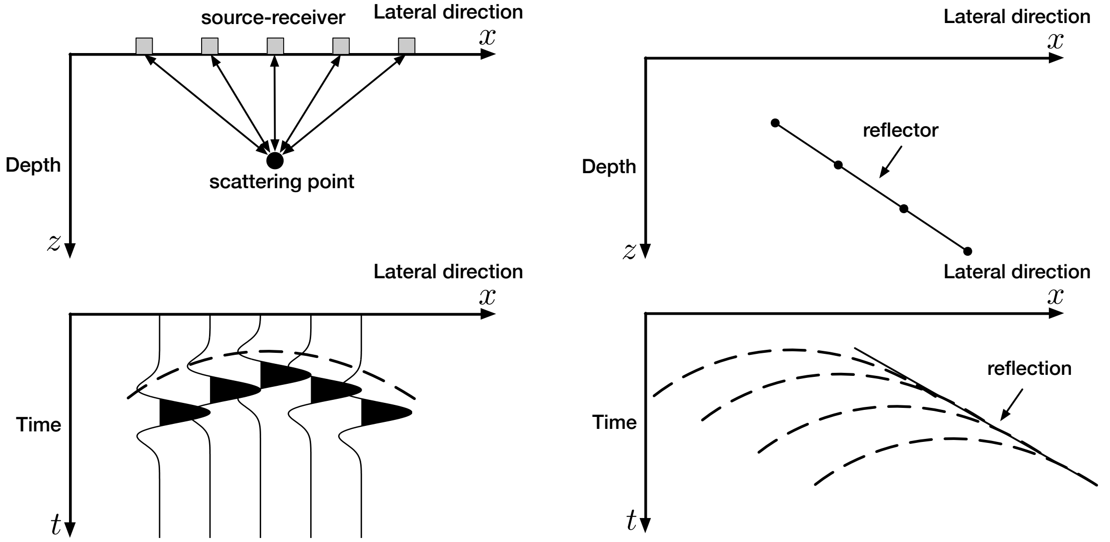


Figure 1.1: A land seismic survey illustration

1.1.2 Seismic Migration and Modeling

Seismic waves that propagate through the earth are governed approximately by the acoustic, elastic and viscous properties of the rock in which they are traveling. When they propagate through an interface between two rock types with different densities and seismic velocities, seismic energy is either reflected, refracted or attenuated. The reflected seismic energy arrives at the surface and is recorded by the receivers. Figure 1.2(a) illustrates a simple homogeneous (constant P-wave velocity) medium with an isolated scattering point at some depth and the corresponding seismic data with a hyperbolic diffraction pattern. While the actual subsurface is far more complicated than that shown in Figure 1.2(a), the seismic data can be represented as a superposition of many diffraction curves generated by each of many point-like anomalies in the subsurface. Figure 1.2(b) illustrates another example with a dipping reflector, where the envelope of many weak diffractions from closely spaced scattering points along the reflector forms a straight reflection line. Note that the reflection is displaced laterally from the true reflector position, and this lateral mispositioning of reflections from dipping reflectors gives rise to the term seismic migration for the process that corrects the positioning. The purpose of seismic migration is to remove distortions from seismic records by moving events to their correct spatial positions and by collapsing energy from diffractions back to their positions at the reflectors [50].



(a) Schematic depth section (top) and seismic data (bottom) for a single scattering point

(b) Schematic depth section (top) and seismic data (bottom) for a dipping reflector

Figure 1.2: Illustrations of seismic migration

The wave equation is an important second-order partial differential equation (PDE) used to model the propagation of seismic waves in a medium, and it serves as the foundation of seismic migration. As a simple example, the 2D acoustic wave equation for a medium with constant density is as follows

$$\left(m(\mathbf{x})\frac{\partial^2}{\partial t^2} - \nabla^2\right) p(\mathbf{x}, t; \mathbf{x}_s) = f(\mathbf{x}, t; \mathbf{x}_s), \tag{1.1}$$

where $\mathbf{x} \triangleq (x, z)$ is the 2D Cartesian coordinates in which x is the lateral coordinate and z is the vertical coordinate, $m(\mathbf{x}) \triangleq \dfrac{1}{v^2(\mathbf{x})}$ is the model parameter, i.e., squared slowness, of position $\mathbf{x}$ given that $v(\mathbf{x})$ is the acoustic wave velocity, $\nabla^2 \triangleq \dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial z^2}$ is the 2D Laplace operator, $p(\mathbf{x}, t; \mathbf{x}_s)$ is the acoustic pressure wavefield as a function of position $\mathbf{x}$ and time t, parameterized by the source position $\mathbf{x}_s$, $f(\mathbf{x}, t; \mathbf{x}_s)$ is the source excitation function generated at position $\mathbf{x}_s$ and $f(\mathbf{x}, t; \mathbf{x}_s) = f(t)\delta(\mathbf{x} - \mathbf{x}_s)$ for a point source. The PDE (1.1) can be solved both forward and backward in time.

In the analysis of seismic migration, it is appropriate to assume $m(\mathbf{x})$ is a background incident model that is sufficiently smooth on the scale of a wavelength. Denoting the reflector as a small perturbation $\delta m(\mathbf{x})$ imposed on the background model $m(\mathbf{x})$, the scattered wavefield $\delta p(\mathbf{x}, t; \mathbf{x}_s)$ satisfies the following PDE based on the Born approximation theory [51, 139] (see Appendix B for details):

$$\left(m(\mathbf{x})\frac{\partial^2}{\partial t^2} - \nabla^2\right) \delta p(\mathbf{x}, t; \mathbf{x}_s) = -\delta m(\mathbf{x})\frac{\partial^2 p}{\partial t^2}(\mathbf{x}, t; \mathbf{x}_s). \tag{1.2}$$

Unlike (1.1) in which the relationship between $m(\mathbf{x})$ and $p(\mathbf{x}, t; \mathbf{x}_s)$ is nonlinear, (1.2) maps $\delta m(\mathbf{x})$ to $\delta p(\mathbf{x}, t; \mathbf{x}_s)$ in a linear way. This is because the background wavefield $p(\mathbf{x}, t; \mathbf{x}_s)$ is determined by the background model $m(\mathbf{x})$, hence can be regarded as fixed for the purpose of determining the scattered wavefield. Therefore, if one compactly denotes (1.1) as a nonlinear forward modeling process from the model $\mathbf{m} \triangleq \{m(\mathbf{x})\}$ to the data $\mathbf{d} \triangleq \{p(\mathbf{x}_r, t; \mathbf{x}_s)\}$ sampled at receiver locations $\mathbf{x}_r$ by using the nonlinear operator $\boldsymbol{\mathcal{F}}(\cdot)$

$$\mathbf{d} \triangleq \boldsymbol{\mathcal{F}}(\mathbf{m}), \tag{1.3}$$

then its Born approximation can also be compactly denoted as a linearized forward modeling process from the reflector $\delta\mathbf{m} \triangleq \{\delta m(\mathbf{x})\}$ to the scattered data

$\delta\mathbf{d} \triangleq \{\delta p(\mathbf{x}_r, t; \mathbf{x}_s)\}$ by using the Jacobian matrix $\mathbf{J} \triangleq \dfrac{\partial \boldsymbol{\mathcal{F}}}{\partial \mathbf{m}}$

$$\delta\mathbf{d} \triangleq \mathbf{J}\delta\mathbf{m}. \tag{1.4}$$

Given the recorded data denoted by $\mathbf{d}_{\text{obs}}$, reverse time migration (RTM) [6, 86, 138] is a preferred approach for estimating $\delta\mathbf{m}$ even if it is structurally complex. It uses the adjoint linear operator $\mathbf{J}^\dagger$ that maps the recorded scattered data $\mathbf{d}_{\text{obs}} - \boldsymbol{\mathcal{F}}(\mathbf{m})$ to the model space

$$\delta\mathbf{m}_{\text{RTM}} = \mathbf{J}^\dagger \left(\mathbf{d}_{\text{obs}} - \boldsymbol{\mathcal{F}}(\mathbf{m})\right). \tag{1.5}$$

In many RTM implementations, amplitudes of the reflectors are ignored and considered fairly unreliable. However, the preservation of the amplitudes becomes a main concern for modern seismic migration algorithms. Least-squares reverse time migration (LSRTM) [114] is able to generate amplitude-preserved imaging results by minimizing the linear least-squares misfit function

$$J(\delta\mathbf{m}) \triangleq \frac{1}{2} \left\|\left(\mathbf{d}_{\text{obs}} - \boldsymbol{\mathcal{F}}(\mathbf{m})\right) - \mathbf{J}\delta\mathbf{m}\right\|_2^2, \tag{1.6}$$

whose closed-form solution is

$$\delta\mathbf{m}_{\text{LSRTM}} = \left(\mathbf{J}^\dagger\mathbf{J}\right)^{-1} \mathbf{J}^\dagger \left(\mathbf{d}_{\text{obs}} - \boldsymbol{\mathcal{F}}(\mathbf{m})\right). \tag{1.7}$$

Comparing LSRTM (1.7) with RTM (1.5), it is obvious that LSRTM applies the preconditioning matrix $\left(\mathbf{J}^\dagger\mathbf{J}\right)^{-1}$ to the RTM result. Since $\mathbf{J}^\dagger\mathbf{J}$ is usually hard to compute and invert, most LSRTM implementations minimize (1.6) by using iterative gradient descent algorithms [37, 125] where the RTM result (1.5) can be regarded as the first iteration of LSRTM.

1.1.3 Full Waveform Inversion

The purpose of full waveform inversion (FWI) [65, 127] is to recover the model $\mathbf{m}$ by fitting the forward modeling data $\boldsymbol{\mathcal{F}}(\mathbf{m})$ to the recorded data $\mathbf{d}_{\text{obs}}$ in a comprehensive way, such that not only the scattering waves caused by reflectors but all information in waveforms, such as travel times, amplitudes, converted waves, multiples, etc., are accounted for. Using a conceptually similar idea with LSRTM, FWI searches for a best estimate to $\mathbf{m}$ by minimizing the following least-squares misfit function

$$E(\mathbf{m}) \triangleq \frac{1}{2}\|\mathbf{d}_{\text{obs}} - \boldsymbol{\mathcal{F}}(\mathbf{m})\|_2^2. \tag{1.8}$$

FWI searches the minimum of $E(\mathbf{m})$ in an iterative manner $\mathbf{m}_{k+1} = \mathbf{m}_k + \delta\mathbf{m}_k$, $k = 0, 1, 2, \ldots$ where $\delta\mathbf{m}_k$ is the optimal descent direction (a.k.a. model perturbation) that minimizes $E(\mathbf{m})$ in the vicinity of the current model $\mathbf{m}_k$. Hence, one can expand $E(\mathbf{m})$ in a small vicinity $\delta\mathbf{m}$ of $\mathbf{m}_k$ with a Taylor polynomial of degree two

$$E(\mathbf{m}) = E(\mathbf{m}_k) + \delta\mathbf{m}^T\mathbf{g}_k + \frac{1}{2}\delta\mathbf{m}^T\mathbf{H}_k\delta\mathbf{m} + o(\|\delta\mathbf{m}\|^3) \tag{1.9}$$

where $\mathbf{g}_k \triangleq \dfrac{\partial E(\mathbf{m}_k)}{\partial\mathbf{m}}$ denotes the gradient of the misfit function $E(\mathbf{m})$ evaluated at $\mathbf{m}_k$ and $\mathbf{H}_k \triangleq \dfrac{\partial^2 E(\mathbf{m}_k)}{\partial\mathbf{m}^2}$ denotes the Hessian matrix whose elements are the second-order partial derivatives of $E(\mathbf{m})$ at $\mathbf{m}_k$. In each iteration, by letting the gradient of $E(\mathbf{m})$ expressed in (1.9) with respect to $\delta\mathbf{m}$ be zero, $\delta\mathbf{m}_k$ satisfies

$$\mathbf{H}_k\delta\mathbf{m}_k = -\mathbf{g}_k. \tag{1.10}$$

After calculating the gradient $\mathbf{g}_k$ and the Hessian matrix $\mathbf{H}_k$, one is able to determine the optimal model perturbation $\delta\mathbf{m}_k$ at each FWI iteration. Since the relationship between the data and the model is nonlinear in FWI, many iterations are required to make the misfit function converge toward a minimum. A schematic workflow of FWI is depicted in Figure 1.3.

1.1.4 Numerical Wave Modeling

Many numerical schemes have been proposed for modeling seismic wave propagation based on PDEs like (1.1). Explicit finite-difference time-domain (FDTD) methods were originally developed in the early 1970's [4, 61], and have been widely used in

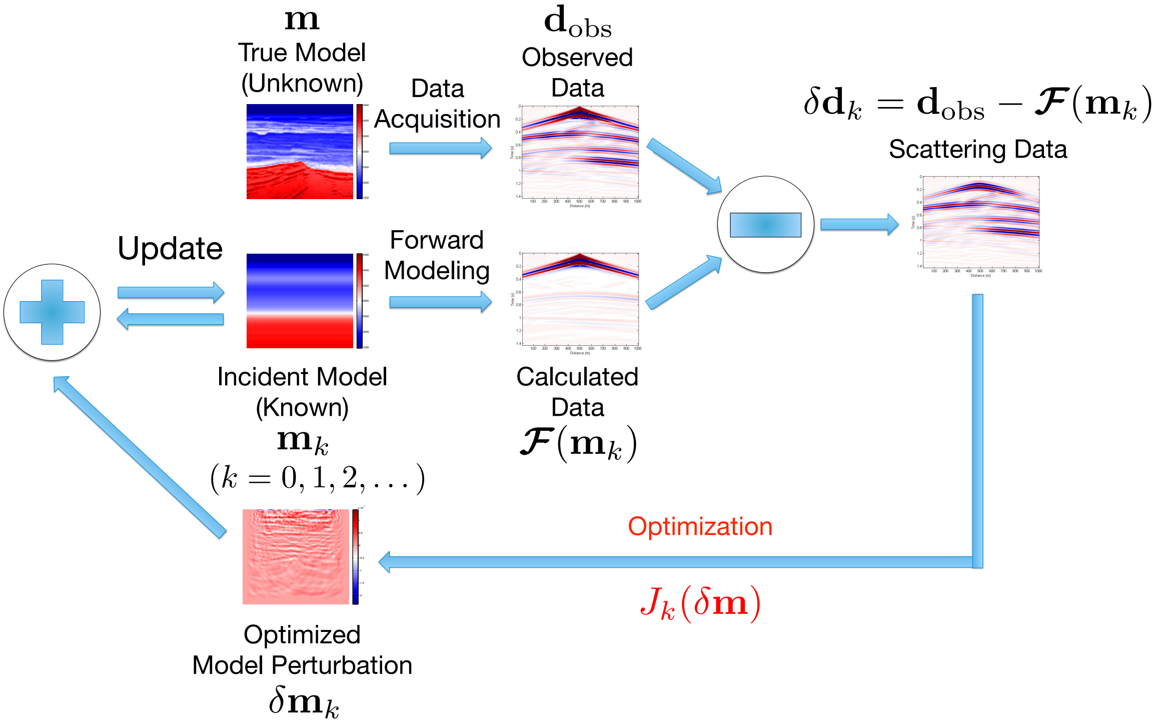


Figure 1.3: Schematic FWI Workflow

both research and industry. Since the early 1990's, finite-difference frequency-domain (FDFD) methods [2, 103, 104] have been actively applied for seismic modeling. The prerequisite of FDFD methods is to have the frequency-domain wave equations in hand. For example, by applying the temporal Fourier transform to the acoustic wave equation (1.1) with a point source, its frequency-domain equivalent is

$$\left(-m(\mathbf{x})\omega^2 - \nabla^2\right) \hat{p}(\mathbf{x}; \omega, \mathbf{x}_s) = \hat{f}(\omega)\delta(\mathbf{x} - \mathbf{x}_s), \tag{1.11}$$

where $\omega = 2\pi f$ is the (angular) frequency parameter, $\hat{f}(\omega)$ is the Fourier transform of $f(t)$ and $\hat{p}(\mathbf{x}; \omega, \mathbf{x}_s)$ is the frequency-domain acoustic pressure wavefield generated by a monochromatic point source term $\hat{f}(\omega)\delta(\mathbf{x} - \mathbf{x}_s)$. FDFD methods discretize $\mathbf{x}$ in (1.11) like FDTD methods discretize $\mathbf{x}$ and t in (1.1).

The FDTD methods are intuitive and easy to understand. As the name suggests, since it is a time-domain technique, when a broadband pulse is used as the source function, the wavefield over a wide range of frequencies can be obtained with a single simulation. However, FDTD requires having a small discretized time step to satisfy the Courant-Friedrichs-Lewy condition [30] for stability. Therefore, a long-time simulation of wave propagation would lead to very huge computational cost.

The FDFD methods have some major differences with the FDTD counterparts. FDFD is easier to implement because there are no time steps that need to be computed sequentially, hence FDFD is able to pick only a proportion of frequencies to compute, leading to a smaller data space dimension. FDFD reduces the frequency-domain wave equation (1.11) to a system of linear equations that can be compactly written as

$$\mathbf{B}(\mathbf{m},\omega)\hat{\mathbf{p}}(\omega;\mathbf{x}_s) = \hat{\mathbf{f}}(\omega;\mathbf{x}_s) \tag{1.12}$$

where $\mathbf{B}(\mathbf{m},\omega)$ is the impedance matrix [85] that is square, non-symmetric, sparse and complex-valued and is characterized by the model $\mathbf{m}$ and the frequency ω, and the column vectors $\hat{\mathbf{p}}(\omega;\mathbf{x}_s)$, $\hat{\mathbf{f}}(\omega;\mathbf{x}_s)$ collect all $\hat{p}(\mathbf{x};\omega,\mathbf{x}_s)$, $\hat{f}(\omega)\delta(\mathbf{x}-\mathbf{x}_s)$ as entries, respectively.

Since each frequency ω is independent of each other in (1.11), one would solve $\hat{p}(\mathbf{x};\omega,\mathbf{x}_s)$ with multiple source $\hat{f}(\omega)\delta(\mathbf{x}-\mathbf{x}_s)$ at multiple frequencies simultaneously, using parallel computing if possible.

1.2 Sparse Signal Processing

Many different classes of signals, such as images, videos, seismic datasets and velocity models are compressible, and can be well approximated by a linear combination of only a few atoms from an appropriate dictionary. Consider a discrete signal $\mathbf{y} \in \mathbb{R}^N$, which can be approximated as a linear combination of unique vectors $\mathbf{d}_i \in \mathbb{R}^N$, $i = 1,\dots,L$, and $N \leq L$, as

$$\mathbf{y} = \sum_{i=1}^{L} x_i \mathbf{d}_i + \mathbf{n} \tag{1.13}$$

where $\mathbf{x} \triangleq [x_1,\dots,x_L]^T \in \mathbb{R}^L$ is the coefficient vector, and $\mathbf{n} \in \mathbb{R}^N$ is the approximation error. Each vector $\mathbf{d}_i$ is called an atom and the set of all atoms $\mathbf{D} \triangleq \{\mathbf{d}_1,\dots,\mathbf{d}_L\}$

is called a dictionary. An atom is used interchangeably as a column vector of $\mathbf{D} \triangleq [\mathbf{d}_1, \ldots, \mathbf{d}_L] \in \mathbb{R}^{N \times L}$, and if $\mathbf{D}$ is an explicit matrix, then (1.13) can be written as $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{n}$. Any signal $\mathbf{y}$ is "sparse" or "compressible" over $\mathbf{D}$ if only $K \ll L$ entries in $\mathbf{x}$ are nonzero values while the remaining $(L - K)$ values are zero. This concept can be mathematically described as $\|\mathbf{x}\|_0 \triangleq \#\{i : x_i \neq 0, i = 1, \ldots, L\} = K$ where the ℓ_0-norm $\| \cdot \|_0$ counts the nonzero entries.

Research into designing good dictionaries $\mathbf{D}$ for different families of signals has never rested. It is always appealing to look for the sparsest coefficient vector $\mathbf{x}$ that solves

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon \tag{1.14}$$

Available dictionary design techniques fall into two categories. The first category assumes a specific type of signal regularity that can be constructed by an analytic model. This generally leads to model-based transforms with implicit dictionaries that are described by structured algorithms, and exemplified by wavelets [84], ridgelets [36], curvelets [20, 21, 25], contourlets [34], seislets [49], etc. They have already been widely used for seismic data processing [53, 54, 73, 78, 117, 140, 143]. The second category infers a dictionary from a set of examples by directly placing sparsity constraints on the coefficients. This sort of data-driven dictionary is written as an explicit matrix, and is better able to adapt to nonintuitive signal regularities beyond piecewise smoothness.

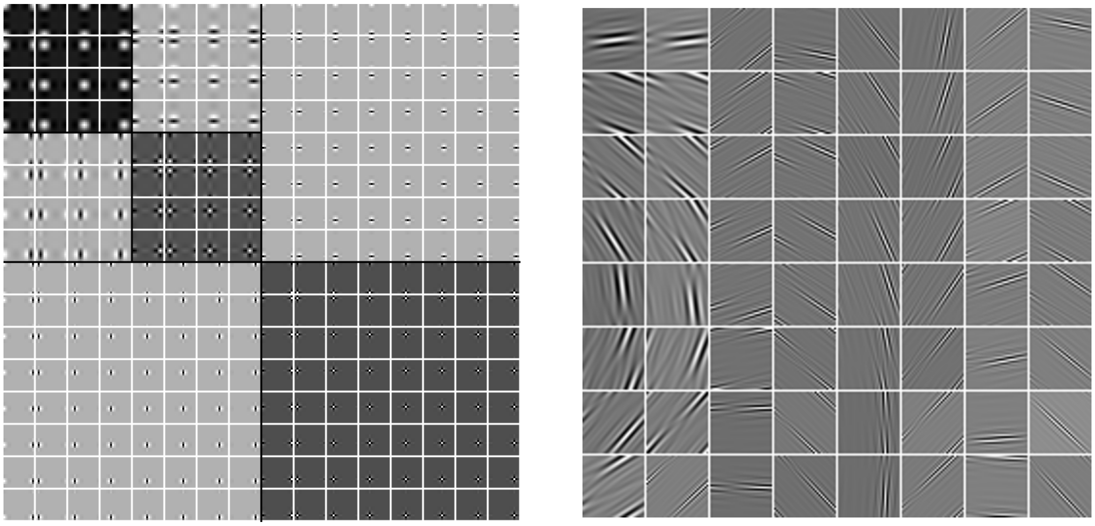
1.2.1 Multi-scale Transforms: From Wavelets to Curvelets

Multi-scale transforms are model-driven processes based on a top-down strategy. They design fix-shaped filters (mother atoms) to capture multi-dimensional signals with assumed features such as scan-lines and smooth curves with sparse coefficients. Hence their success in applications relies on how well the signals fit the assumptions. These transforms have efficient algorithmic implementations in the spatial-frequency domain and, as a result, their representations as dictionaries $\mathbf{D}$ are implicit.

The most basic discrete multi-scale transform is the discrete wavelet transform [32], whose dictionary $\mathbf{D} = \left\{ \psi_{j,k}(t) \triangleq \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - k \cdot 2^j}{2^j}\right) \middle| j, k \in \mathbb{Z} \right\}$ is a set of dyadic scaled (by a factor of 2^j) and shifted (by a factor of $k \cdot 2^j$) versions of the mother wavelet $\psi(t)$. A signal $\mathbf{y}$ can be uniquely represented in a wavelet expansion

$$\mathbf{y} = \sum_j \sum_k x_{j,k} \boldsymbol{\psi}_{j,k} \tag{1.15}$$

where the wavelet coefficients are $x_{j,k} = \langle \mathbf{y}, \boldsymbol{\psi}_{j,k} \rangle$. Practically, low- and high-pass filter banks are used to implement wavelet atoms. Figure 1.4(a) shows a 2D wavelet dictionary that is a Kronecker product of two 1D wavelet dictionaries across three scales, and whose atoms are isotropic and optimal to horizontal, vertical and diagonal scan-lines in the sense of approximation error [84].



(a) Wavelets

(b) Curvelets

Figure 1.4: Wavelet and curvelet atoms

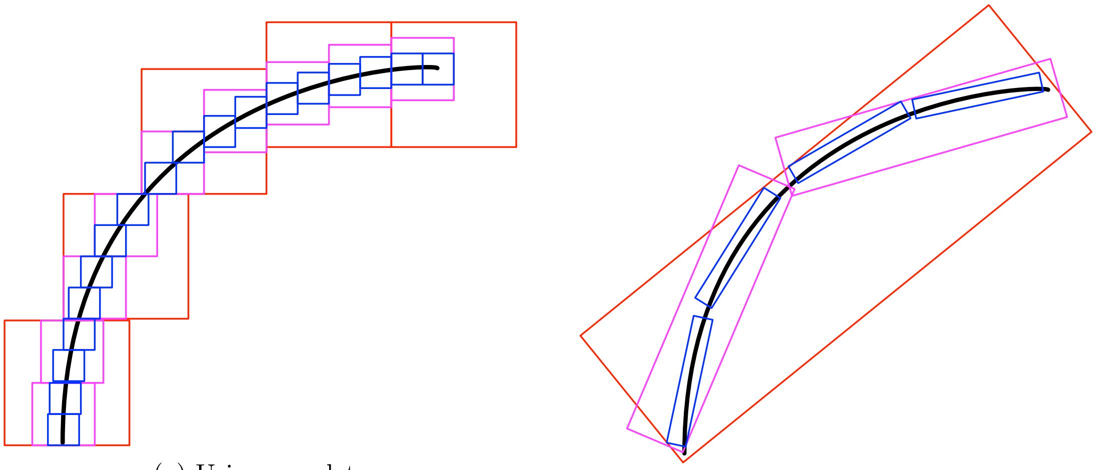
Most natural signals such as images exhibit piecewise smooth curves. However, a 2D discrete wavelet transform cannot represent curves with sparse coefficients since its atoms lack directional selectivity. The curvelet transform [20, 21, 23, 24, 33] was proposed to overcome this problem, and it allows an optimal sparse representation of piecewise smooth curves with respect to approximation error [20, 21]. The curvelet dictionary can be written as

$$\mathbf{D} = \left\{ \psi_{j,l,\mathbf{k}}(\mathbf{u}) \triangleq \psi_j \left(\boldsymbol{\mathcal{R}}_{\theta_l} \left(\mathbf{u} - \mathbf{u}_{\mathbf{k}}^{(j,l)} \right) \right) \middle| j, l \in \mathbb{Z}, \theta_l = 2\pi \cdot 2^{-\lfloor j/2 \rfloor} \cdot l, \mathbf{k} \triangleq (k_1, k_2) \in \mathbb{Z}^2 \right\}$$

where $\psi_j(\mathbf{u})$ is the dilated mother curvelet which is a function of the 2D coordinate $\mathbf{u}$ whose frequency support is a band-pass wedge, $\boldsymbol{\mathcal{R}}_\theta$ is the rotation operator by θ radians such that $\psi_j(\boldsymbol{\mathcal{R}}_\theta(\mathbf{u}))$ has an oriented trapezoid window in the frequency domain with angle θ, and $\mathbf{u}_{\mathbf{k}}^{(j,l)} \triangleq \boldsymbol{\mathcal{R}}_{\theta_l}^{-1} \left(2^{-j} \cdot k_1, 2^{-j/2} \cdot k_2 \right)$ to make sure the curvelets have a parabolic scaling relation: width $\approx$ length^2. Given this curvelet dictionary, a 2D signal $\mathbf{y}$ can be represented in a curvelet expansion

$$\mathbf{y} = \sum_j \sum_l \sum_{\mathbf{k}} x_{j,l,\mathbf{k}} \boldsymbol{\psi}_{j,l,\mathbf{k}} \tag{1.16}$$

where the curvelet coefficients are $x_{j,l,\mathbf{k}} = \langle \mathbf{y}, \boldsymbol{\psi}_{j,l,\mathbf{k}} \rangle$. Compared to wavelets, curvelets have one more parameter l that controls the direction of the atom. Figure 1.4(b) shows some curvelets from a dictionary across three scales, in which one can see these needle-shaped atoms are anisotropic and possess very high directional selectivity.



(a) Using wavelets

(b) Using curvelets

Figure 1.5: Approximation of a curve using wavelet and curvelet atoms

Figure 1.5 illustrates the approximation of a curve with multi-scale wavelets and curvelets. Wavelets are isotropic and their frequency-domain supports are fixed-area windows for one scale, so that capturing a curve requires many coefficients in different scales, as shown in Figure 1.5(a). On the contrary, curvelets are anisotropic and their frequency-domain supports are directional windows with parabolic scaling. Therefore, representing a curve requires only a few coefficients per scale, as shown in Figure 1.5(b).

1.2.2 Dictionary Learning

Dictionary learning, unlike the top-down design strategy of multi-scale transforms, is a data-driven process that infers the dictionary $\mathbf{D} \in \mathbb{R}^{N \times L}$, $N \leq L$, from a set of training examples. In this case, $\mathbf{D}$ is typically an explicit matrix that yields the sparsest representations for the training examples. Learning $\mathbf{D}$ is a bottom-up machine learning strategy by enforcing sparsity constraints on the coefficients and adapting the elements of $\mathbf{D}$ to the training examples.

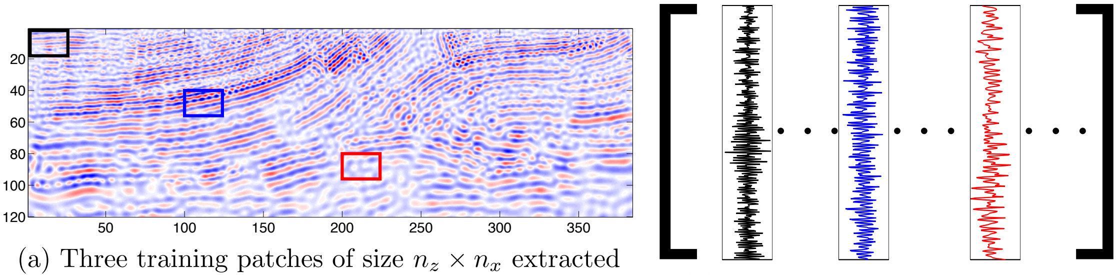
A probabilistic framework is used in the development of dictionary learning [71, 95, 96]. It starts with the sparse approximation model that represents an arbitrary signal $\mathbf{y} \in \mathbb{R}^N$ as

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{n} \tag{1.17}$$

where $\mathbf{x} \in \mathbb{R}^L$ is a sparse coefficient vector and $\mathbf{n} \in \mathbb{R}^N$ is Gaussian noise.

It is worth noting that though $\mathbf{y}$ is always a column vector in this context, it does not refer only to a 1D signal. For example, a 2D image patch of size $n_z \times n_x$ where n_z and n_x is the height and width of the patch, respectively, is equivalent to a vector $\mathbf{y}$ of length $N = n_z n_x$ after reshaping, and reshaping can convert any multi-dimensional signal to a 1D vector $\mathbf{y} \in \mathbb{R}^N$. The terms patch and its reshaped vector can be used interchangeably. For example, Figure 1.6 illustrates three 2D image patches as well as their reshaped column vectors from a model perturbation image.

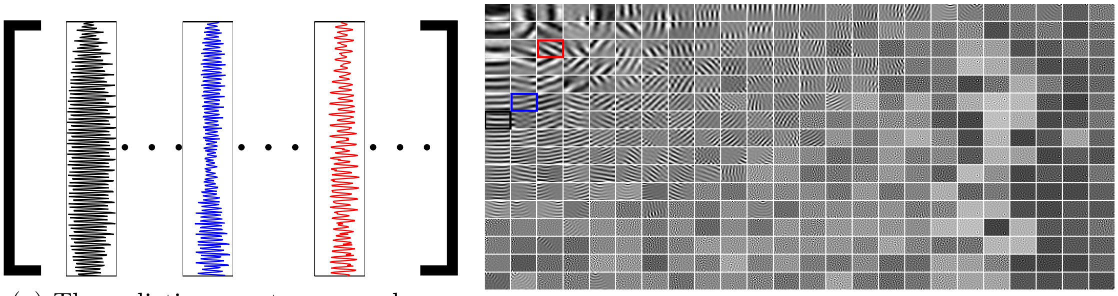
For patch-based dictionary learning, it is also a convention to present the dictionary matrix $\mathbf{D}$ by reshaping each dictionary atom of length $N = n_z n_x$ back to a block of size $n_z \times n_x$ for better visualization of their geometric features, as shown in Figure 1.7. Hence one can illustrate atoms of $\mathbf{D}$ as blocks. The sparse approximation $\mathbf{y} \approx \mathbf{D}\mathbf{x}$ is illustrated in Figure 1.8 with vectors or patch blocks interchangeably.



(a) Three training patches of size $n_z \times n_x$ extracted from a model perturbation

(b) Three training patches that are reshaped into column vectors of length $N = n_z n_x$

Figure 1.6: Examples of training patches vectorized into columns



(a) Three dictionary atoms as column vectors of length $N = n_z n_x$ in $\mathbf{D}$

(b) All dictionary atoms in $\mathbf{D}$ are reshaped into 2D blocks of size $n_z \times n_x$ for better visualization

Figure 1.7: Dictionary atoms are reshaped into blocks for better visualization.

Given a matrix of R training examples $\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_R] \in \mathbb{R}^{N \times R}$, the dictionary learning method seeks the dictionary matrix $\hat{\mathbf{D}}$ that maximizes the likelihood function $P(\mathbf{Y}|\mathbf{D})$:

$$\hat{\mathbf{D}} = \underset{\mathbf{D}}{\operatorname{argmax}} \, P(\mathbf{Y}|\mathbf{D}). \tag{1.18}$$

With the assumption that each example $\mathbf{y}_i$, $i = 1, \ldots, R$, is drawn independently,

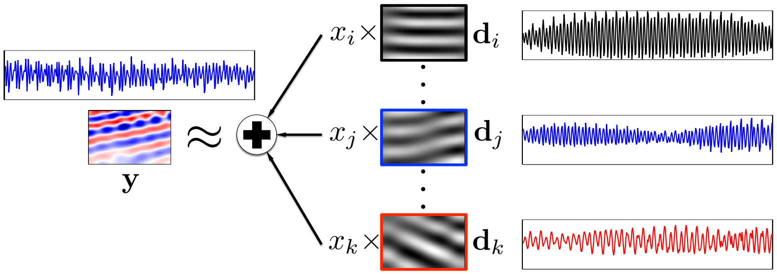


Figure 1.8: Approximation of the patch $\mathbf{y}$ by a few dictionary atoms is written as a matrix-vector product $\mathbf{y} \approx \mathbf{D}\mathbf{x} = \sum_i x_i \mathbf{d}_i$, but is equivalent to summing a few atoms from the dictionary visualized in Figure 1.7.

the likelihood function is

$$P(\mathbf{Y}|\mathbf{D}) = \prod_{i=1}^{R} P(\mathbf{y}_i|\mathbf{D}). \tag{1.19}$$

Thus the maximum likelihood (ML) expression of (1.18) becomes

$$\hat{\mathbf{D}} = \underset{\mathbf{D}}{\operatorname{argmax}} \prod_{i=1}^{R} P(\mathbf{y}_i|\mathbf{D}) = \underset{\mathbf{D}}{\operatorname{argmax}} \sum_{i=1}^{R} \log\left(P(\mathbf{y}_i|\mathbf{D})\right) \tag{1.20}$$

where $P(\mathbf{y}_i|\mathbf{D})$ can be expressed in terms of its coefficients $\mathbf{x}_i$ as

$$P(\mathbf{y}_i|\mathbf{D}) = \int_{\mathbf{x}_i} P(\mathbf{y}_i, \mathbf{x}_i|\mathbf{D}) \mathrm{d}\mathbf{x}_i = \int_{\mathbf{x}_i} P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{D}) P(\mathbf{x}_i) \mathrm{d}\mathbf{x}_i. \tag{1.21}$$

Because the analytic solution of this integration is difficult to solve, Olshausen and Field [95] handled this by replacing the integral with the maximum, which gives

$$\hat{\mathbf{D}} = \underset{\mathbf{D}}{\operatorname{argmax}} \sum_{i=1}^{R} \max_{\mathbf{x}_i} \left\{\log\left(P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{D}) P(\mathbf{x}_i)\right)\right\}. \tag{1.22}$$

Since the coefficient vector $\mathbf{x}_i$ is sparse, each element could be assumed to be a zero-mean, independent and identically distributed (i.i.d.) Laplacian random variable with scale $1/\mu$. Also, each element of the noise term $\mathbf{n}$ could be assumed to be a zero-mean i.i.d. Gaussian random variable with σ^2 as variance. Then,

$$\begin{aligned} P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{D}) &= C_g \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2\right) \\ P(\mathbf{x}_i) &= C_l \exp\left(-\mu\|\mathbf{x}_i\|_1\right), \end{aligned} \tag{1.23}$$

where C_g and C_l are normalization constants.

After inserting (1.23) into (1.22), the overall ML estimation problem for learning the dictionary $\mathbf{D}$ becomes

$$\hat{\mathbf{D}} = \underset{\mathbf{D}}{\operatorname{argmin}} \sum_{i=1}^{R} \min_{\mathbf{x}_i} \left\{ \frac{1}{2\sigma^2} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2 + \mu \|\mathbf{x}_i\|_1 \right\}. \tag{1.24}$$

This problem can be solved with an iterative alternating optimization scheme, which first finds the sparsity coefficients $\{\mathbf{x}_i\}$ given a fixed dictionary $\mathbf{D}$ and then updates the elements of $\mathbf{D}$ with the known and fixed sparse coefficients $\{\mathbf{x}_i\}$.

The probabilistic framework leads to many successful dictionary learning algorithms, including the K-singular value decomposition (K-SVD) [1], the method of optimal directions (MOD) [45], generalized principal component analysis (GPCA) [134], orthonormal dictionary learning [115, 116], unions of orthonormal bases [69], and others.

1.2.3 Compressive Sensing

The sparse approximation of a signal $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{n}$ with the condition that $\|\mathbf{x}\|_0 = K$ motivates the idea of compressive sensing (CS), proposed by Candès, Romberg, Tao [22] and Donoho [39], to acquire a much smaller number of measurements for processing such signals more efficiently as compared to the classical Nyquist-Shannon sampling theorem. Instead of using $\mathbf{y} \in \mathbb{R}^N$ by sampling at twice the bandwidth of its continuous signal, CS suggests measuring another signal $\mathbf{z} \in \mathbb{R}^M$ through a subsampling matrix $\mathbf{W} \in \mathbb{R}^{M \times N}$ $(M \leq N)$ as

$$\mathbf{z} = \mathbf{W}\mathbf{y} = \mathbf{W}\mathbf{D}\mathbf{x} + \mathbf{W}\mathbf{n} = \boldsymbol{\Theta}\mathbf{x} + \boldsymbol{\eta} \tag{1.25}$$

where $\boldsymbol{\Theta} \triangleq \mathbf{W}\mathbf{D} \in \mathbb{R}^{M \times L}$ is the measurement matrix and $\boldsymbol{\eta} \triangleq \mathbf{W}\mathbf{n}$ is the measurement error. The design of $\boldsymbol{\Theta}$ must allow stable reconstruction of $\mathbf{y} \in \mathbb{R}^N$ from $\mathbf{z} \in \mathbb{R}^M$. A tractable necessary condition for stable reconstruction of $\mathbf{y}$ relies on the mutual coherence of $\boldsymbol{\Theta}$ [38, 40, 41]

$$\mu(\boldsymbol{\Theta}) = \max_{i \neq j} \frac{|\boldsymbol{\theta}_i^H \boldsymbol{\theta}_j|}{\|\boldsymbol{\theta}_i\|_2 \|\boldsymbol{\theta}_j\|_2}. \tag{1.26}$$

The condition $M \geq \mu(\mathbf{\Theta})^2 K \log N$ guarantees recovery of $\mathbf{y}$ from $\mathbf{z}$ with high probability [108]. Proper subsampling matrices $\mathbf{W}$ that come with small mutual coherence $\mu(\mathbf{\Theta})$ include i.i.d. Gaussian random matrices [39], random convolution Toeplitz matrices [108] and randomly selected rows from identity matrices [55, 80], etc.

The reconstruction of $\mathbf{y}$ is equivalent to finding the sparsest coefficients $\mathbf{x}$ that satisfy (1.25), which can be formulated as follows

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{z} - \mathbf{\Theta x}\|_2 \leq \epsilon \tag{1.27}$$

where ϵ is an estimate of the reconstruction error. This problem can be reformulated into two different forms. The first one is

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{\Theta x}\|_2 + \mu \|\mathbf{x}\|_0, \tag{1.28}$$

where μ is the Lagrange multiplier that tunes the trade-off between the approximation error and sparsity constraint. Another one is

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{\Theta x}\|_2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq \tau \tag{1.29}$$

where τ is an estimate of the sparsity level.

Unfortunately, solving optimization problems such as (1.27), (1.28) and (1.29) with ℓ_0-norm constraints is generally NP-hard for arbitrary $\mathbf{\Theta}$ [90]. In practical, greedy strategies such as matching pursuit (MP) [83], orthogonal matching pursuit (OMP) [99, 132] and their variants have been developed to approximately solve these problems. Another workaround is to replace the ℓ_0-norm $\|\mathbf{x}\|_0$ with the ℓ_1-norm $\|\mathbf{x}\|_1 = \sum_{i=1}^{L} |x_i|$ [39], yielding the following three problems that can be solved by convex programming methods

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{z} - \mathbf{\Theta x}\|_2 \leq \epsilon, \tag{1.30}$$

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{\Theta x}\|_2 + \mu \|\mathbf{x}\|_1, \tag{1.31}$$

and

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{z} - \boldsymbol{\Theta}\mathbf{x}\|_2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq \tau. \tag{1.32}$$

Among them, (1.30) is defined as basis pursuit denoising (BPDN) problem [28] and (1.32) has the name of least absolute shrinkage and selection operator (LASSO) [130, 131]. These formulations are equivalent as long as the parameters ϵ, μ and τ are appropriately selected [133] and such an equivalence relation plays an important role in sparse signal processing problems.

1.3 Thesis Organization

The objective of this thesis is to improve the quality of the recorded seismic data and the efficiency of seismic inversion algorithms so as to deliver high-fidelity earth models that can be used for oil and gas reservoir exploration and characterization. The thesis is organized as follows.

Chapter 2 proposes a novel seismic data reconstruction scheme using a double-sparsity dictionary learning method. Section 2.2 introduces the K-SVD algorithm that is a renowned implementation of dictionary learning. Section 2.3 investigates the structure of the adaptive dictionary learned from the seismic data and proposes a double-sparsity dictionary learning method in which the learned dictionary $\mathbf{D}$ is constructed as a multiplication of a base dictionary $\boldsymbol{\Phi}$ corresponding to a fixed analytic transform with a sparse matrix $\mathbf{A}$ that actually needs to be learned. Such a cascaded form for the learned dictionary strikes a good balance among complexity, adaptivity, and performance. Section 2.4 describes the overall seismic dataset denoising scheme as a fused iterative procedure that comprises signal denoising and dictionary update. Section 2.5 extends the discussion of dictionary learning to the case that the noise is nonhomogeneous, or more specifically, entire traces are missing. Section 2.6 presents some experimental results that demonstrate the effectiveness of the algorithms.

Chapter 3 proposes a novel CS-based framework for the FWI problem using sparsity promotion based on orthonormal dictionary learning. Section 3.2 reviews the FWI problem in the frequency domain by iteratively updating the model perturbation via the Gauss-Newton method, and analyzes its high data dimensionality and intensive computational complexity in detail. Section 3.3 introduces the orthonormal dictionary learning as a fast and efficient algorithm and then proposes an adaptive transform called the Sparse Orthonormal Transform (SOT) based on the learned orthonormal dictionary for representing the entire model perturbation. In order to match the iterative property of FWI, an online approach for orthonormal dictionary learning is also proposed in this section, where the dictionary is continually updated by using training patches extracted from the model perturbations in previous iterations, so that the sequence of learned dictionaries can adapt to the variations of patches in later iterations and the extra learning overhead is greatly reduced. Section 3.4 proposes the CS-based framework for FWI by coupling both compressive subsampling and the adaptive SOT-based sparse representation into the Gauss-Newton least-squares problem for each FWI iteration. The result shows that the model perturbation can be well recovered after an ℓ_1-norm sparsity constraint is applied on the SOT coefficients even when only a small proportional of seismic data is used for inversion. Section 3.5 presents some numerical experiments on velocity models to demonstrate that the SOT-based sparsity promoting regularization can provide robust FWI results with greatly reduced computation.

Chapter 4 summarizes the main contributions of the thesis and discusses some potential future extensions for this line of research.

Appendix A presents a flexible and scalable software package with the name *Seismic Simulation, Survey, and Imaging (SSSI)* that was developed along with the thesis for seismic simulations and inversion. It was developed in both MATLAB® and C, and provides basic building blocks for seismic inversion such as numerical wave modeling by finite difference methods (FDTD and FDFD), the construction of Green's functions and the Jacobian matrix, etc. Some large-scale matrix-vector multiplications are overloaded by efficient matrix-free functional operations. Parallel computing is extensively implemented in the software to accelerate processing. This software can be downloaded from the website of The Center for Energy and Geo Processing (CeGP) at `http://cegp.ece.gatech.edu`. Appendix B gives a brief review of the Born approximation that is indispensable for the seismic inversion algorithms.

CHAPTER II

SEISMIC DATA RECOVERY THROUGH SPARSITY-PROMOTING DICTIONARY LEARNING

2.1 Introduction

Seismic data quality is vital to geophysical applications. However, a seismic dataset is often contaminated by random and ambient noise sources, such as ground roll, reverberating refractions, equipment malfunctions, etc. In addition, real data acquisition may encounter different sorts of obstacles, such as buildings, highways, fences, etc. These obstacles, coupled with limited recording capacity or greater cost, result in missing or nonuniform spatial traces. Noisy and missing traces will hamper the ability to obtain reliable subsurface images, making seismic data reconstruction a critical step in seismic data processing flows prior to seismic imaging.

Seismic data can be reconstructed in a transform domain where signal sparsity is exploited, e.g., wavelet [27, 143], contourlet [118], or curvelet [53, 54, 92] transforms. These transforms assume specific types of regularities within signals and build analytical, and thus fixed, multi-scale bases for sparse representation. Transform-domain methods are efficient and treat the seismic dataset as a whole volume. However, this may not be the best strategy when the seismic dataset exhibits repetitive localized features of wave fronts. Alternatively, dictionary learning methods that infer explicit and adaptive dictionary matrices from patch-based training sets can also be used to reconstruct corrupted seismic data [7, 126, 142]. These methods offer refined dictionaries that adapt to the localized features of the data under processing and yield much better performance in many applications. However, one disadvantage of dictionary learning is high overhead including the need to store explicit dictionary matrices.

This chapter [145] proposes a seismic data reconstruction scheme based on a novel dictionary learning method called double-sparsity dictionary learning. This method is motivated by a hypothesis that the learned dictionary atoms themselves may be represented by sparse coefficients over another more fundamental dictionary and suggests forming the overall dictionary as a multiplication of a fixed transform and a sparse matrix. Such a cascaded form of the learned dictionary combines the efficiency from a fixed transform with the adaptability from dictionary learning.

The rest of this chapter is organized as follows. Section 2.2 reviews the renowned K-SVD algorithm for dictionary learning. Section 2.3 introduces the motivation of dictionary model with double-sparsity constraints and the details of the sparse K-SVD algorithm. Section 2.4 describes the patch-based seismic data denoising using the learned double-sparsity dictionary. Learning separate multi-scale dictionaries and performing denoising in different subbands of a multi-scale transform are also presented in this section. Section 2.5 extends the method to seismic dataset inpainting where many traces are missing. Section 2.6 gives the numerical experiments of denoising and inpainting.

2.2 The K-SVD Algorithm

Given a training set $\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_R] \in \mathbb{R}^{N \times R}$ in which each element is a column vector of length N, the goal of dictionary learning is to find a matrix $\mathbf{D} \in \mathbb{R}^{N \times L}$, $N \leq L$, that is able to represent $\mathbf{Y}$ with a set of sparse coefficients summarized as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_R] \in \mathbb{R}^{L \times R}$. This is an ML estimation problem with respect to $\mathbf{D}$ which has been discussed in Chapter 1:

$$\hat{\mathbf{D}} = \operatorname*{argmin}_{\mathbf{D}} \sum_{i=1}^{R} \min_{\mathbf{x}_i} \left\{ \frac{1}{2\sigma^2} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2 + \mu \|\mathbf{x}_i\|_1 \right\}. \qquad \text{(1.24 revisited)}$$

This problem does not have any constraint on the elements of the dictionary matrix $\mathbf{D}$ as it does for those of coefficients $\mathbf{x}_i$; thus, the naïve solution of (1.24) tends to increase the element values of $\mathbf{D}$ in order to allow those of $\mathbf{x}_i$ to become as small as possible. This issue can be handled by constraining each atom of $\mathbf{D}$ to be normalized to one in the ℓ_2-norm so that the element values of $\mathbf{x}_i$ are kept at an appropriate level [96]. Then, the dictionary learning problem becomes

$$\left\{\hat{\mathbf{D}}, \hat{\mathbf{X}}\right\} = \underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \mu\|\mathbf{X}\|_1 \quad \text{subject to} \quad \begin{cases} \|\mathbf{d}_j\|_2 = 1 \\ \quad \forall j = 1, \ldots, L \end{cases} \tag{2.1}$$

where $\|\cdot\|_F$ is the Frobenius norm.

The K-SVD algorithm is able to solve (2.1) by using an iterative strategy that alternates between two steps in which each step reduces (2.1) into a problem that involves only one unknown by fixing another one as known. The first step finds the sparse coefficients in $\mathbf{X}$ of all input training patches in $\mathbf{Y}$ with the current dictionary estimate $\mathbf{D}$, so that (2.1) is reduced into

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \mu\|\mathbf{X}\|_1. \tag{2.2}$$

According to the equivalence relation among (1.30), (1.31) and (1.32), the above problem can be decoupled into R distinct LASSO problems for sparse representations of training examples $\mathbf{y}_i$, $\forall i = 1, \ldots, R$, over the fixed dictionary $\mathbf{D}$ as

$$\mathbf{x}_i = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}\|_F^2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq t, \quad \forall\, i = 1, \ldots, R \tag{2.3}$$

where t is the ℓ_1-norm sparsity level of each coefficient vector $\mathbf{x}$.

The second step updates the current dictionary estimate $\mathbf{D}$ with the known sparse coefficients $\mathbf{X}$ found in the first step, and reduces (2.1) into

$$\hat{\mathbf{D}} = \underset{\mathbf{D}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{subject to} \quad \begin{cases} \|\mathbf{d}_j\|_2 = 1 \\ \quad \forall j = 1, \ldots, L. \end{cases} \tag{2.4}$$

This problem can be solved by updating one atom $\mathbf{d}_j$ at a time, while preserving the ℓ_1-norm sparsity constraints on $\mathbf{x}_i$. In order to achieve this, updating the atom $\mathbf{d}_j$ only takes those training examples in $\mathbf{Y}$ whose sparse representations in $\mathbf{X}$ have used

$\mathbf{d}_j$ into account. The column index set of these training examples in $\mathbf{Y}$ that have used $\mathbf{d}_j$ for sparse representation is denoted by $\mathcal{I}_j$ and can be obtained by locating nonzero elements in the j-th row of $\mathbf{X}$, i.e.,

$$\mathcal{I}_j \triangleq \{r | 1 \leq r \leq R, x_{jr} \neq 0\}, \tag{2.5}$$

so that all coefficients corresponding to $\mathbf{d}_j$ can be denoted by $\mathbf{X}_{j,\mathcal{I}_j}$, which is a row vector that takes elements with index $\mathcal{I}_j$ from the j-th row of $\mathbf{X}$ and is updated along with $\mathbf{d}_j$ as well.

The objective function in (2.4) that considers only the columns with index $\mathcal{I}_j$ from $\mathbf{Y}$ can be written as

$$\begin{aligned} \left\| \mathbf{Y}_{\mathcal{I}_j} - \mathbf{D}\mathbf{X}_{\mathcal{I}_j} \right\|_F^2 &= \left\| \left(\mathbf{Y}_{\mathcal{I}_j} - \sum_{i \neq j} \mathbf{d}_i \mathbf{X}_{i,\mathcal{I}_j} \right) - \mathbf{d}_j \mathbf{X}_{j,\mathcal{I}_j} \right\|_F^2 \\ &= \left\| \mathbf{E}_j - \mathbf{d}_j \mathbf{X}_{j,\mathcal{I}_j} \right\|_F^2 \end{aligned} \tag{2.6}$$

where $\mathbf{E}_j \triangleq \mathbf{Y}_{\mathcal{I}_j} - \sum\limits_{i \neq j} \mathbf{d}_i \mathbf{X}_{i,\mathcal{I}_j}$ is the residual matrix if the atom $\mathbf{d}_j$ is removed. Therefore, the resulting problem for updating the atom $\mathbf{d}_j$ as well as its corresponding coefficients $\mathbf{X}_{j,\mathcal{I}_j}$ becomes

$$\left\{ \mathbf{d}_j, \mathbf{X}_{j,\mathcal{I}_j}^T \right\} = \underset{\mathbf{d},\mathbf{x}}{\operatorname{argmin}} \left\| \mathbf{E}_j - \mathbf{d}\mathbf{x}^T \right\|_F^2 \quad \text{subject to} \quad \|\mathbf{d}\|_2 = 1. \tag{2.7}$$

This is a simple rank-1 matrix approximation problem and hence can be directly solved by the SVD of $\mathbf{E}_j$. If the SVD of $\mathbf{E}_j$ is denoted by $\mathbf{E}_j = \mathbf{U\Sigma V}^T$, then the atom $\mathbf{d}_j$ is updated by the first column of $\mathbf{U}$, and the updated corresponding coefficients $\mathbf{X}_{j,\mathcal{I}_j}^T$ are updated by the first column of $\mathbf{V}$ multiplied by the first diagonal element (i.e., the largest singular value) of $\mathbf{\Sigma}$, which leads to

$$\mathbf{E}_j = \mathbf{U\Sigma V}^T \quad \Rightarrow \quad \begin{cases} \mathbf{d}_j = \mathbf{u}_1 \\ \mathbf{X}_{j,\mathcal{I}_j}^T = \sigma_{11} \mathbf{v}_1. \end{cases} \tag{2.8}$$

The detailed implementation of the K-SVD algorithm is summarized in Algorithm 2.1.

```
Input: Training set $\mathbf{Y} \in \mathbb{R}^{N \times R}$, signal sparsity level $t$, and number of training
       iterations $K$
Output: Learned dictionary $\mathbf{D} \in \mathbb{R}^{N \times L}$, sparse coefficient matrix $\mathbf{X} \in \mathbb{R}^{L \times R}$
Initialization: $\mathbf{D} \leftarrow \mathbf{D}_0$ (a pre-chosen matrix), $\mathbf{X} \leftarrow \mathbf{0}$
repeat
    for $i \leftarrow 1$ to $R$ do
        // Sparse coding
        $\mathbf{x}_i \leftarrow \underset{\mathbf{x}}{\text{argmin}} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}\|_2^2$   subject to   $\|\mathbf{x}\|_1 \leq t$;
    end
    for $j \leftarrow 1$ to $L$ do
        $\mathcal{I}_j \leftarrow \{r | 1 \leq r \leq R, x_{jr} \neq 0\}$;
        // Atom removal
        $\mathbf{d}_j \leftarrow \mathbf{0}$;
        $\mathbf{E}_j \leftarrow \mathbf{Y}_{\mathcal{I}_j} - \mathbf{D}\mathbf{X}_{\mathcal{I}_j}$;
        // Atom updating
        $\mathbf{E}_j \leftarrow \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ ; // Compute SVD
        $\mathbf{d}_j \leftarrow \mathbf{u}_1$;
        $\mathbf{X}_{j,\mathcal{I}_j}^T \leftarrow \sigma_{11}\mathbf{v}_1$;
    end
until $K$ training iterations;
```

Algorithm 2.1: The K-SVD algorithm

For applications in image processing, any attempt to directly use full-size images for dictionary learning would yield intractable computational complexity. Instead, one should feed the dictionary learning algorithm with small image patches of size $n_z \times n_x$ such that $N = n_z n_x$ is of moderate size. Besides the complexity issue, small patches exhibit better local self-similarities than large patches and, therefore, the resulting dictionary can have better representation ability on local features. Such locality can be turned back into a global treatment of full-size images by appropriately tiling the small patches, which will be described later.

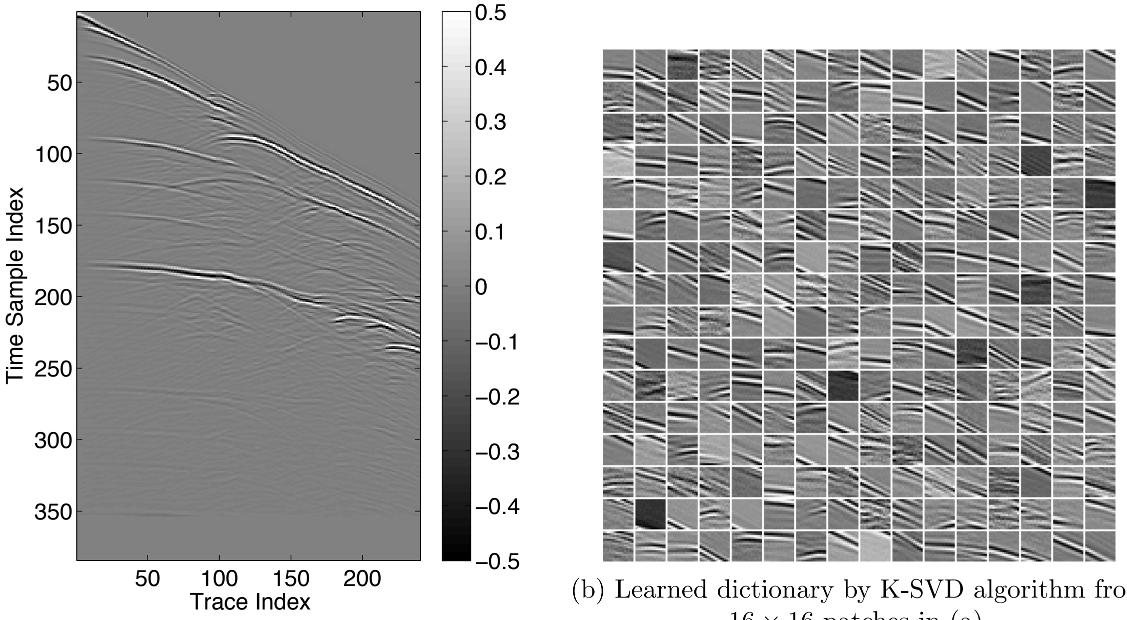
2.3 Double-Sparsity Dictionary Learning

The K-SVD algorithm [1] trains the dictionary $\mathbf{D} \in \mathbb{R}^{N \times L}$ that is adapted to a training set $\mathbf{Y} \in \mathbb{R}^{N \times R}$ by solving the dictionary learning problem (2.1). It has been widely used to handle various image processing and computer vision tasks, such as denoising [44, 68], inpainting [72, 82, 119], super-resolution [100, 141], etc.

Compared with fixed multi-scale dictionaries, the learned dictionaries are more adaptive to the data and produce better results. However, these gains do not come for free. Dictionary learning algorithms based on the iterative alternating optimization approach bring extra computational overhead. Since the learned dictionaries are explicit matrices, extra space is required to store their elements, and applying signal reconstruction by matrix-vector multiplication would be less efficient than applying multi-scale transforms with fast algorithms. Furthermore, no prior structural information is involved in the construction of the dictionary, yet this would not always be the case. In the seismic application, datasets have distinct structural patterns, which can help to guide the design of dictionaries.

What would be an appropriate dictionary to represent seismic data? A seismic dataset is a collection of data traces, each one of which is a recorded continuous waveform from a seismic source. Many traces together provide a spatio-temporal sampling of the wavefield, which contains wave fronts along straight lines and hyperbolae that correspond to direct ray paths and reflections with normal moveouts, respectively. This structural information can be exploited to improve dictionary learning.

As an example, Figure 2.1(b) demonstrates an example of a learned dictionary with $L = 256$ atoms obtained by the K-SVD algorithm on a set of 16×16 patches ($N = 256$) from a synthetic seismic dataset shown in Figure 2.1(a). Though there are no constraints posed by the algorithm, we can notice the strong resemblance among atoms in the resulting dictionary, which suggests that the atoms themselves may share some underlying structures that can be represented over a more fundamental base dictionary with sparsity.



(a) Synthetic seismic dataset

(b) Learned dictionary by K-SVD algorithm from 16×16 patches in (a).

Figure 2.1: Synthetic data and a learned dictionary

2.3.1 A Dictionary Model with Double Sparsity

The resemblance among atoms in the dictionary obtained by the K-SVD algorithm suggests that atoms can have sparse representations over some base dictionary. This concept is called “double-sparsity”, which was first proposed in the image processing literature [111].

The double-sparsity dictionary model can be described as

$$\mathbf{D} = \mathbf{\Phi}\mathbf{A} \tag{2.9}$$

where $\mathbf{\Phi} \in \mathbb{R}^{N\times L}$ is the base dictionary generally chosen to have a quick implicit implementation and $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_L] \in \mathbb{R}^{L\times L}$ is a sparse matrix to be learned in which each column satisfies $\|\mathbf{a}_i\|_1 \leq p$ for some sparsity level p. Therefore, the double-sparsity dictionary can be regarded as a two-level cascaded combination of two dictionaries. Usually, the base dictionary $\mathbf{\Phi}$ is selected as a synthesis operator of some fixed transform that comes with a fast implementation. The sparse matrix $\mathbf{A}$ can be regarded as an extension to the existing analytic transform, adding a new layer of adaptivity on the fixed $\mathbf{\Phi}$. Comparing with the regular unstructured dictionary $\mathbf{D}$ which is a fully explicit matrix, the double-sparsity dictionary model (2.9) is significantly more efficient because only a few elements in $\mathbf{A}$ need to be learned, stored and transmitted. More importantly, due to its fewer degrees of freedom, such a dictionary model reduces the chance of overfitting the noise in the training set and produces robust results even with limited training examples. These properties are particularly advantageous for the process of denoising and inpainting of seismic datasets.

By inserting the double-sparsity dictionary model (2.9) into (2.1), the double-sparsity dictionary learning optimization problem is formulated as

$$\left\{\hat{\mathbf{A}}, \hat{\mathbf{X}}\right\} = \underset{\mathbf{A},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{\Phi}\mathbf{A}\mathbf{X}\|_F^2 + \mu\|\mathbf{X}\|_1 \quad \text{subject to} \quad \begin{cases} \|\mathbf{a}_j\|_1 \leq p \\ \|\mathbf{\Phi}\mathbf{a}_j\|_2 = 1 \\ \forall j = 1, \ldots, L \end{cases} \tag{2.10}$$

2.3.2 The Sparse K-SVD Algorithm

The sparse K-SVD algorithm is a dictionary learning algorithm specifically designed to learn the sparse dictionary $\mathbf{A} \in \mathbb{R}^{L\times L}$ by solving the optimization problem (2.10). It is a variant of the K-SVD algorithm and hence inherits its basic strategy of iteratively alternating between sparse representation and dictionary update.

Algorithm 2.2 presents the sparse K-SVD algorithm with details. Similar to the K-SVD algorithm, in each learning iteration, the first step decouples (2.10) into R distinct LASSO problems as

$$\mathbf{x}_i = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y}_i - \mathbf{\Phi}\mathbf{A}\mathbf{x}\|_F^2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq t, \quad \forall\, i = 1, \ldots, R \tag{2.11}$$

and determines the sparse representation $\mathbf{x}_i$ of each training example $\mathbf{y}_i$ with the current dictionary $\mathbf{D} = \mathbf{\Phi}\mathbf{A}$ fixed.

Different from the K-SVD algorithm, the second step in each iteration updates each sparse column $\mathbf{a}_j$ of the matrix $\mathbf{A}$ to formulate the renewed atom $\mathbf{d}_j = \mathbf{\Phi}\mathbf{a}_j$. Still, by denoting a column index set $\mathcal{I}_j$ of the training examples in $\mathbf{Y}$ whose representations use $\mathbf{d}_j$ as (2.5), the objective function to update $\mathbf{a}_j$ is equivalent to

$$\begin{aligned} \left\|\mathbf{Y}_{\mathcal{I}_j} - \mathbf{\Phi}\mathbf{A}\mathbf{X}_{\mathcal{I}_j}\right\|_F^2 &= \left\|\left(\mathbf{Y}_{\mathcal{I}_j} - \sum_{i\neq j} \mathbf{\Phi}\mathbf{a}_i\mathbf{X}_{i,\mathcal{I}_j}\right) - \mathbf{\Phi}\mathbf{a}_j\mathbf{X}_{j,\mathcal{I}_j}\right\|_F^2 \\ &= \left\|\mathbf{E}_j - \mathbf{\Phi}\mathbf{a}_j\mathbf{X}_{j,\mathcal{I}_j}\right\|_F^2 \end{aligned} \tag{2.12}$$

where $\mathbf{E}_j = \mathbf{Y}_{\mathcal{I}_j} - \sum\limits_{i\neq j} \mathbf{\Phi}\mathbf{a}_i\mathbf{X}_{i,\mathcal{I}_j}$ is the residual matrix without the contribution of $\mathbf{d}_j$. Therefore, the resulting problem to update $\mathbf{a}_j$ and $\mathbf{X}_{j,\mathcal{I}_j}$ is given by

$$\left\{\mathbf{a}_j, \mathbf{X}_{j,\mathcal{I}_j}^T\right\} = \underset{\mathbf{a},\mathbf{x}}{\operatorname{argmin}} \left\|\mathbf{E}_j - \mathbf{\Phi}\mathbf{a}\mathbf{x}^T\right\|_F^2 \quad \text{subject to} \quad \begin{cases} \|\mathbf{a}\|_1 \leq p \\ \|\mathbf{\Phi}\mathbf{a}\|_2 = 1 \end{cases} \tag{2.13}$$

Unlike solving (2.7) in the K-SVD algorithm, (2.13) cannot be solved simply as a rank-1 matrix approximation problem with SVD operations. Instead, [11] proposed an alternative method, which guarantees a reduction of the objective function. Suppose $\mathbf{x}$ in (2.13) is fixed and the norm constraint $\|\mathbf{\Phi}\mathbf{a}\|_2 = 1$ is temporarily put aside, $\mathbf{a}_j$ can be optimized by solving the following problem

$$\mathbf{a}_j = \underset{\mathbf{a}}{\operatorname{argmin}} \left\|\mathbf{E}_j - \mathbf{\Phi}\mathbf{a}\mathbf{x}^T\right\|_F^2 \quad \text{subject to} \quad \|\mathbf{a}\|_1 \leq p. \tag{2.14}$$

However, this problem is difficult to solve. The following theorem provided in [111] shows that (2.14) can be converted into a much simpler problem.

Theorem 2.1 *Let $\mathbf{E} \in \mathbb{R}^{N\times M}$, $\mathbf{\Phi} \in \mathbb{R}^{N\times L}$ be two matrices, and $\mathbf{a} \in \mathbb{R}^L$, $\mathbf{x} \in \mathbb{R}^M$ be two vectors, and also let $\|\mathbf{x}\|_2 = 1$, then the following equation holds*

$$\left\|\mathbf{E} - \mathbf{\Phi}\mathbf{a}\mathbf{x}^T\right\|_F^2 = \|\mathbf{E}\mathbf{x} - \mathbf{\Phi}\mathbf{a}\|_2^2 + f(\mathbf{E}, \mathbf{x}),$$

where $f(\mathbf{E}, \mathbf{x})$ is not a function of $\mathbf{a}$.

Proof The left-hand side can be expanded as

$$\begin{aligned}
\left\|\mathbf{E}-\boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T\right\|_F^2 &= \mathrm{Tr}\left((\mathbf{E}-\boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T)^T(\mathbf{E}-\boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T)\right) \\
&= \mathrm{Tr}\left(\mathbf{E}^T\mathbf{E}\right) - 2\mathrm{Tr}\left(\mathbf{E}^T\boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T\right) + \mathrm{Tr}\left(\mathbf{x}\mathbf{a}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T\right) \\
&= \mathrm{Tr}\left(\mathbf{E}^T\mathbf{E}\right) - 2\mathrm{Tr}\left(\mathbf{x}^T\mathbf{E}^T\boldsymbol{\Phi}\mathbf{a}\right) + \mathrm{Tr}\left(\mathbf{x}^T\mathbf{x}\mathbf{a}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{a}\right) \\
&= \mathrm{Tr}\left(\mathbf{E}^T\mathbf{E}\right) - 2\mathbf{x}^T\mathbf{E}^T\boldsymbol{\Phi}\mathbf{a} + \mathbf{a}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{a}.
\end{aligned}$$

The right-hand side can be expanded as

$$\begin{aligned}
\|\mathbf{E}\mathbf{x}-\boldsymbol{\Phi}\mathbf{a}\|_2^2 &= (\mathbf{E}\mathbf{x}-\boldsymbol{\Phi}\mathbf{a})^T(\mathbf{E}\mathbf{x}-\boldsymbol{\Phi}\mathbf{a}) \\
&= \mathbf{x}^T\mathbf{E}^T\mathbf{E}\mathbf{x} - 2\mathbf{x}^T\mathbf{E}^T\boldsymbol{\Phi}\mathbf{a} + \mathbf{a}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{a} \\
\therefore \left\|\mathbf{E}-\boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T\right\|_F^2 &= \|\mathbf{E}\mathbf{x}-\boldsymbol{\Phi}\mathbf{a}\|_2^2 + \mathrm{Tr}\left(\mathbf{E}^T\mathbf{E}\right) - \mathbf{x}^T\mathbf{E}^T\mathbf{E}\mathbf{x} \\
&= \|\mathbf{E}\mathbf{x}-\boldsymbol{\Phi}\mathbf{a}\|_2^2 + f(\mathbf{E},\mathbf{x}),
\end{aligned}$$

where $f(\mathbf{E},\mathbf{x}) \triangleq \mathrm{Tr}\left(\mathbf{E}^T\mathbf{E}\right) - \mathbf{x}^T\mathbf{E}^T\mathbf{E}\mathbf{x}$. ▮

Theorem 2.1 suggests that $\mathbf{a}_j$ can be optimized by solving the following sparse representation problem of $\mathbf{E}_j\mathbf{x}$ over $\boldsymbol{\Phi}$ after normalizing $\mathbf{x} = \mathbf{X}_{j,\mathcal{I}_j}^T/\|\mathbf{X}_{j,\mathcal{I}_j}^T\|_2$

$$\mathbf{a}_j = \underset{\mathbf{a}}{\mathrm{argmin}}\,\|\mathbf{E}_j\mathbf{x}-\boldsymbol{\Phi}\mathbf{a}\|_F^2 \quad \text{subject to} \quad \|\mathbf{a}\|_1 \le p. \tag{2.15}$$

After obtaining $\mathbf{a}_j$, $\mathbf{X}_{j,\mathcal{I}_j}^T$ can be solved by

$$\mathbf{X}_{j,\mathcal{I}_j}^T = \underset{\mathbf{x}}{\mathrm{argmin}}\,\|\mathbf{E}_j-\boldsymbol{\Phi}\mathbf{a}_j\mathbf{x}^T\|_F^2 \tag{2.16}$$

This problem can be solved by matrix calculus. According to the left-hand side expansion in Theorem 2.1, the gradient of $\|\mathbf{E}_j-\boldsymbol{\Phi}\mathbf{a}_j\mathbf{x}^T\|_F^2$ with respect to $\mathbf{x}$ is

$$\begin{aligned}
\frac{\partial\|\mathbf{E}_j-\boldsymbol{\Phi}\mathbf{a}_j\mathbf{x}^T\|_F^2}{\partial\mathbf{x}} &= \frac{\partial\mathrm{Tr}\left(\mathbf{x}^T\mathbf{x}\mathbf{a}_j^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{a}_j\right)}{\partial\mathbf{x}} - 2\frac{\partial\mathbf{x}^T\mathbf{E}_j^T\boldsymbol{\Phi}\mathbf{a}_j}{\partial\mathbf{x}} \\
&= 2\mathbf{a}_j^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{a}_j\mathbf{x} - 2\mathbf{E}_j^T\boldsymbol{\Phi}\mathbf{a}_j
\end{aligned}$$

Then the solution $\mathbf{X}_{j,\mathcal{I}_j}^T$ sets the gradient equal to zero and has the form

$$\mathbf{X}_{j,\mathcal{I}_j}^T = \frac{\mathbf{E}_j^T\boldsymbol{\Phi}\mathbf{a}_j}{\|\boldsymbol{\Phi}\mathbf{a}_j\|_2}, \tag{2.17}$$

where the dictionary atom $\boldsymbol{\Phi}\mathbf{a}_j$ has been effectively normalized to unit length.

```
Input: Training set $\mathbf{Y} \in \mathbb{R}^{N \times R}$, base dictionary $\mathbf{\Phi} \in \mathbb{R}^{N \times L}$, signal sparsity
       level $t$, atom sparsity level $p$, and number of training iterations $K$
Output: Sparse dictionary $\mathbf{A} \in \mathbb{R}^{L \times L}$, sparse coefficient matrix $\mathbf{X} \in \mathbb{R}^{L \times R}$
Initialization: $\mathbf{A} \leftarrow \mathbf{I}$, $\mathbf{X} \leftarrow \mathbf{0}$
repeat
    for $i \leftarrow 1$ to $R$ do
        // Sparse coding
        $\mathbf{x}_i \leftarrow \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y}_i - \mathbf{\Phi A x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq t$;
    end
    for $j \leftarrow 1$ to $L$ do
        $\mathcal{I}_j \leftarrow \{r | 1 \leq r \leq R, x_{jr} \neq 0\}$;
        $\mathbf{x} \leftarrow \mathbf{X}_{j,\mathcal{I}_j}^T / \|\mathbf{X}_{j,\mathcal{I}_j}^T\|_2$;
        // Atom removal
        $\mathbf{a}_j \leftarrow \mathbf{0}$;
        $\mathbf{r} \leftarrow \mathbf{Y}_{\mathcal{I}_j}\mathbf{x} - \mathbf{\Phi A X}_{\mathcal{I}_j}\mathbf{x}$;
        // Atom updating
        $\mathbf{a}_j \leftarrow \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{r} - \mathbf{\Phi a}\|_2^2 \quad \text{subject to} \quad \|\mathbf{a}\|_1 \leq p$;
        $\mathbf{a}_j \leftarrow \mathbf{a}_j / \|\mathbf{\Phi a}_j\|_2$;
        $\mathbf{X}_{j,\mathcal{I}_j} \leftarrow \left(\mathbf{Y}_{\mathcal{I}_j}^T \mathbf{\Phi a} - \left(\mathbf{\Phi A X}_{\mathcal{I}_j}\right)^T \mathbf{\Phi a}\right)^T$;
    end
    for $j \leftarrow 1$ to $L$ do
        Atom_Replacing($\mathbf{\Phi a}_j$);
    end
until $K$ training iterations;
```

Algorithm 2.2: Sparse K-SVD algorithm

2.3.3 Atom Replacing Techniques

In the sparse K-SVD algorithm all the dictionary atoms are presumed to be of equal importance, although some ill-posed atoms should be replaced according to certain criteria which will be described below. Such procedures can effectively avoid local minima or overfitting and, therefore, improve the adaptability of the learned dictionary.

The representation ability of the learned dictionary will be reduced if some atoms happen to be very similar. Mutual coherence defined in (1.26) is a useful measurement of the similarity among the atoms in a dictionary matrix $\mathbf{D}$

$$\mu(\mathbf{D}) = \max_{i \neq j} \frac{\mathbf{d}_i^T \mathbf{d}_j}{\|\mathbf{d}_i\|_2 \|\mathbf{d}_j\|_2}. \tag{1.26 revisited}$$

After all L columns of the matrix $\mathbf{A}$ have been updated in a training iteration, $\mu(\mathbf{D}) = \mu(\mathbf{\Phi A})$ will be examined. If there is a pair of $(\mathbf{a}_i, \mathbf{a}_j)$ which makes $\mu(\mathbf{D})$ exceed some threshold (say 0.99), one element (say, $\mathbf{a}_j$) should be replaced with the representation of $\mathbf{y}_k$ over $\mathbf{\Phi}$ that satisfies

$$\mathbf{a}_j = \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{y}_k - \mathbf{\Phi a}\|_2^2 \quad \text{subject to} \quad \|\mathbf{a}\|_1 \leq p \tag{2.18}$$

where k refers to the index of the signal in $\mathbf{Y}$ that exhibits the largest approximation error after taking the current sparse matrix $\mathbf{A}$ into account, i.e.,

$$k = \underset{i}{\operatorname{argmax}} \|\mathbf{y}_i - \mathbf{\Phi A x}_i\|_2^2. \tag{2.19}$$

Because the number of training patches R is always much larger than the number of atoms L, such a replacement prevents similar atoms from appearing again.

Besides the replacement of similar atoms, infrequently used atoms are also identified and replaced. As indicated before, the number of nonzero elements in j-th row of $\mathbf{X}$ indicates that how many training signals in $\mathbf{Y}$ use $\mathbf{d}_j$ in their representations. If an atom is used by less than a threshold number (say 4) of training signals, then it is considered “less representative” and can be replaced with another one that represents more training signals. This atom replacement within $\mathbf{A}$ can be done by solving the optimization problem (2.18) as well.

2.4 Seismic Data Denoising with Learned Dictionary

After carrying out the double-sparsity dictionary learning, it is safe to use the dictionary $\mathbf{D} = \mathbf{\Phi A}$ to denoise small seismic data patches based on the assumption that each patch has a sparse representation over the dictionary.

A small patch of noisy seismic data that has n_x traces and n_z time samples per trace can be reshaped into a vector $\mathbf{w}_0 \in \mathbb{R}^N$ where $N = n_z n_x$. Based on the additive noise model, $\mathbf{w}_0$ has the form of

$$\mathbf{w}_0 = \mathbf{s}_0 + \mathbf{n}_0 \tag{2.20}$$

where $\mathbf{s}_0 \in \mathbb{R}^N$ is the unknown denoised seismic data patch to be estimated and $\mathbf{n}_0 \in \mathbb{R}^N$ is a vector of random noise whose elements are $\mathcal{N}(0, \sigma^2)$. Since $\mathbf{w}_0$ is assumed to have a sparse representation over the learned dictionary $\mathbf{D} = \mathbf{\Phi A}$, the goal of denoising is to estimate $\mathbf{s}_0$ as well as the sparse representation $\mathbf{x}_0$ from $\mathbf{w}_0$ by solving the following problem

$$\{\hat{\mathbf{s}}_0, \mathbf{x}_0\} = \underset{\mathbf{s},\mathbf{x}}{\operatorname{argmin}} \|\mathbf{s} - \mathbf{\Phi A x}\|_2^2 + \mu\|\mathbf{x}\|_1 + \lambda\|\mathbf{s} - \mathbf{w}_0\|_2^2. \tag{2.21}$$

Besides the sparsity penalty term $\mu\|\mathbf{x}\|_1$, another penalty term $\lambda\|\mathbf{s} - \mathbf{w}_0\|_2^2$ controls the proximity between the noisy measurement $\mathbf{w}_0$ and its denoising estimate $\mathbf{s}$.

There are still two unknowns in (2.21) when the dictionary $\mathbf{D} = \mathbf{\Phi A}$ is already known. Similar to the approach that is used in the K-SVD and sparse K-SVD algorithms, (2.21) can be iteratively solved by decoupling it into two alternating optimization steps, each solving one unknown while keeping the other one fixed.

Algorithm 2.3 describes the iterative two-step alternating optimization process of denoising a single seismic data patch with the learned dictionary $\mathbf{D} = \mathbf{\Phi A}$. It starts

```
Input: Learned dictionary $\mathbf{D} = \mathbf{\Phi}\mathbf{A}$, a vectorized noisy seismic data patch
       $\mathbf{w}_0 \in \mathbb{R}^N$, number of denoising iterations $K$
Output: Denoised seismic data patch $\hat{\mathbf{s}}_0$, sparse coefficient vector $\mathbf{x}_0$
Initialization: $\hat{\mathbf{s}}_0 \leftarrow \mathbf{w}_0$
repeat
    $\mathbf{x}_0 \leftarrow \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\hat{\mathbf{s}}_0 - \mathbf{\Phi}\mathbf{A}\mathbf{x}\|_2^2 \leq N\sigma^2$;
    $\hat{\mathbf{s}}_0 \leftarrow (\lambda + 1)^{-1} (\lambda \mathbf{w}_0 + \mathbf{\Phi}\mathbf{A}\mathbf{x}_0)$;
until $K$ denoising iterations;
```

Algorithm 2.3: Denoise a seismic data patch with learned dictionary

with an initialization $\hat{\mathbf{s}}_0 = \mathbf{w}_0$. In each denoising iteration, the first step estimates $\mathbf{x}_0$ given that $\hat{\mathbf{s}}_0$ is fixed, which reduces (2.21) to the problem

$$\mathbf{x}_0 = \underset{\mathbf{x}}{\operatorname{argmin}} \|\hat{\mathbf{s}}_0 - \mathbf{\Phi}\mathbf{A}\mathbf{x}\|_2^2 + \mu\|\mathbf{x}\|_1. \tag{2.22}$$

This step actually seeks the sparse representation of $\hat{\mathbf{s}}_0$ over $\mathbf{D} = \mathbf{\Phi}\mathbf{A}$ as (2.22) can be translated to a BPDN problem

$$\mathbf{x}_0 = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\hat{\mathbf{s}}_0 - \mathbf{\Phi}\mathbf{A}\mathbf{x}\|_2^2 \leq N\sigma^2, \tag{2.23}$$

which can be solved without the need of choosing μ explicitly.

The second step updates the estimator $\hat{\mathbf{s}}_0$ with the obtained $\mathbf{x}_0$, and it reduces (2.21) into a simple least-squares problem

$$\hat{\mathbf{s}}_0 = \underset{\mathbf{s}}{\operatorname{argmin}} \|\mathbf{s} - \mathbf{\Phi}\mathbf{A}\mathbf{x}_0\|_2^2 + \lambda\|\mathbf{s} - \mathbf{w}_0\|_2^2, \tag{2.24}$$

which has a closed-form solution $\hat{\mathbf{s}}_0 = (\lambda + 1)^{-1} (\lambda \mathbf{w}_0 + \mathbf{\Phi}\mathbf{A}\mathbf{x}_0)$.

Denoising small seismic data patches is not a difficult task as many pursuit algorithms can be used to solve (2.22). Now it is time to handle a larger seismic dataset. Let $\mathbf{s} \in \mathbb{R}^{N_z N_x}$ denote an unknown seismic dataset written as a vector that collects N_x traces, each with N_z time samples, where $N_z \gg n_z$ and $N_x \gg n_x$. The denoising process aims to estimate $\mathbf{s}$ from its noisy version $\mathbf{w} = \mathbf{s} + \mathbf{n}$ contaminated by white Gaussian noise $\mathbf{n} \in \mathbb{R}^{N_z N_x}$ with variance σ^2 for each element. It is obviously impossible to denoise $\mathbf{w} \in \mathbb{R}^{N_z N_x}$ by simply replacing $\mathbf{w}_0$ in Algorithm 2.3 with $\mathbf{w}$, because the dictionary matrix $\mathbf{D} = \mathbf{\Phi}\mathbf{A}$ has only $N = n_z n_x$ rows, which is much shorter than the length of $\mathbf{w}$.

In order to denoise the large seismic dataset $\mathbf{w}$ as a whole with the patch-sized dictionary, one must work with small patches of size $n_z \times n_x$ from $\mathbf{w}$ and then tile all denoised results back to form the estimator $\hat{\mathbf{s}}$. By defining an operator $\boldsymbol{\mathcal{R}}_{ij} \in \{0,1\}^{N \times N_z N_x}$ that extracts the (i,j)-th patch of size $n_z \times n_x$ from $\mathbf{w} \in \mathbb{R}^{N_z N_x}$ and reshapes the patch into a vector of length $N = n_z n_x$, $\boldsymbol{\mathcal{R}}_{ij}\mathbf{w} \in \mathbb{R}^N$ is a noisy patch that can be denoised by solving the optimization problem (2.21). Generalizing (2.21) to consider all patches $\boldsymbol{\mathcal{R}}_{ij}\mathbf{w}$, $\forall(i,j)$, the global denoising problem for the entire noisy seismic dataset $\mathbf{w}$ can be formulated as

$$\{\hat{\mathbf{s}}, \mathbf{x}_{ij}\} = \underset{\mathbf{s},\mathbf{x}}{\operatorname{argmin}} \sum_{(i,j)} \|\boldsymbol{\mathcal{R}}_{ij}\mathbf{s} - \mathbf{\Phi}\mathbf{A}\mathbf{x}\|_2^2 + \sum_{(i,j)} \mu_{ij}\|\mathbf{x}\|_1 + \lambda\|\mathbf{s} - \mathbf{w}\|_2^2. \tag{2.25}$$

Similar to the one-patch denoising case, the global denoising problem (2.25) can be solved by an iterative algorithm that alternates between $\hat{\mathbf{s}}$ and $\mathbf{x}_{ij}$. When $\hat{\mathbf{s}}$ is fixed, (2.25) can be decoupled into many smaller BPDN tasks to get $\mathbf{x}_{ij}$, and each one has the form of

$$\mathbf{x}_{ij} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\boldsymbol{\mathcal{R}}_{ij}\hat{\mathbf{s}} - \mathbf{\Phi}\mathbf{A}\mathbf{x}\|_2^2 \leq N\sigma^2, \quad \forall(i,j). \tag{2.26}$$

After all $\mathbf{x}_{ij}$ have been obtained, (2.25) is reduced to the following least-squares problem

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \sum_{(i,j)} \|\boldsymbol{\mathcal{R}}_{ij}\mathbf{s} - \mathbf{\Phi}\mathbf{A}\mathbf{x}_{ij}\|_2^2 + \lambda\|\mathbf{s} - \mathbf{w}\|_2^2, \tag{2.27}$$

yielding the closed-form solution for $\mathbf{s}$ as

$$\hat{\mathbf{s}} = \left(\lambda\mathbf{I} + \sum_{(i,j)} \boldsymbol{\mathcal{R}}_{ij}^\dagger \boldsymbol{\mathcal{R}}_{ij}\right)^{-1} \left(\lambda\mathbf{w} + \sum_{(i,j)} \boldsymbol{\mathcal{R}}_{ij}^\dagger \mathbf{\Phi}\mathbf{A}\mathbf{x}_{ij}\right). \tag{2.28}$$

Since the matrix $\left(\lambda\mathbf{I} + \sum_{(i,j)} \boldsymbol{\mathcal{R}}_{ij}^\dagger \boldsymbol{\mathcal{R}}_{ij}\right)$ is diagonal, this solution can be interpreted as a weighted sum of the tiling result assembled by all reconstructed patches and the original noisy data, followed by a pixel-by-pixel weighted averaging process.

```
Input: Vectorized noisy seismic dataset $\mathbf{w} \in \mathbb{R}^{N_z N_x}$, patch height $n_z$, patch
       width $n_x$, atom size $N = n_z n_x$, base dictionary $\mathbf{\Phi} \in \mathbb{R}^{N \times L}$, number of
       training iterations $K$
Output: Denoised seismic dataset $\hat{\mathbf{s}} \in \mathbb{R}^{N_z N_x}$, sparse matrix $\mathbf{A} \in \mathbb{R}^{L \times L}$, sparse
        coefficient matrix $\mathbf{X} \in \mathbb{R}^{L \times (N_z - n_z + 1)(N_x - n_x + 1)}$
Initialization: $\hat{\mathbf{s}} \leftarrow \mathbf{w}$, $\mathbf{A} \leftarrow \mathbf{I}$, $\mathbf{X} \leftarrow \mathbf{0}$
repeat
    // Sparse Representation Stage
    for $i \leftarrow 1$ to $N_z - n_z + 1$ do
        for $j \leftarrow 1$ to $N_x - n_x + 1$ do
            $\mathbf{x}_{ij} \leftarrow \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\boldsymbol{\mathcal{R}}_{ij}\hat{\mathbf{s}} - \mathbf{\Phi A x}\|_2^2 \le N\sigma^2$;
            Place $\mathbf{x}_{ij}$ into $\mathbf{X}$ as a column with index $(i-1)(N_x - n_x + 1) + j$;
        end
    end
    // Dictionary Update Stage
    for $k \leftarrow 1$ to $L$ do
        $\mathcal{I}_k \leftarrow \{r | 1 \le r \le (N_z - n_z + 1)(N_x - n_x + 1), x_{kr} \neq 0\}$;
        // Atom removal
        $\mathbf{a}_k \leftarrow \mathbf{0}$;
        $\mathbf{x} \leftarrow \mathbf{X}_{k,\mathcal{I}_k}^T / \|\mathbf{X}_{k,\mathcal{I}_k}^T\|_2$;
        $\mathbf{r} \leftarrow \mathbf{Y}_{\mathcal{I}_k}\mathbf{x} - \mathbf{\Phi A X}_{\mathcal{I}_k}\mathbf{x}$;
        // Atom updating
        $\mathbf{a}_k \leftarrow \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{r} - \mathbf{\Phi a}\|_2^2 \quad \text{subject to} \quad \|\mathbf{a}\|_1 \le p$;
        $\mathbf{a}_k \leftarrow \mathbf{a}_k / \|\mathbf{\Phi a}_k\|_2$;
        $\mathbf{X}_{k,\mathcal{I}_k} \leftarrow \left(\mathbf{Y}_{\mathcal{I}_k}^T \mathbf{\Phi a} - (\mathbf{\Phi A X}_{\mathcal{I}_k})^T \mathbf{\Phi a}\right)^T$;
    end
    for $j \leftarrow 1$ to $L$ do
        Atom_Replacing($\mathbf{\Phi a}_k$);
    end
until $K$ training iterations;
// Denoising Stage
$\hat{\mathbf{s}} \leftarrow \left(\lambda \mathbf{I} + \sum_{(i,j)} \boldsymbol{\mathcal{R}}_{ij}^\dagger \boldsymbol{\mathcal{R}}_{ij}\right)^{-1} \left(\lambda \mathbf{w} + \sum_{(i,j)} \boldsymbol{\mathcal{R}}_{ij}^\dagger \mathbf{\Phi A x}_{ij}\right)$
```

Algorithm 2.4: Denoise seismic dataset using the double-sparsity dictionary learned on patches from the noisy dataset

Although the discussion in this section presumes the learned dictionary $\mathbf{D} = \mathbf{\Phi A}$ is fixed, the design of the sparse matrix $\mathbf{A}$ could also be embedded within the denoising process. This can be done by regarding $\mathbf{A}$ in (2.25) also as an unknown, fusing both double-sparsity dictionary learning and seismic dataset denoising together into a new problem with three unknowns

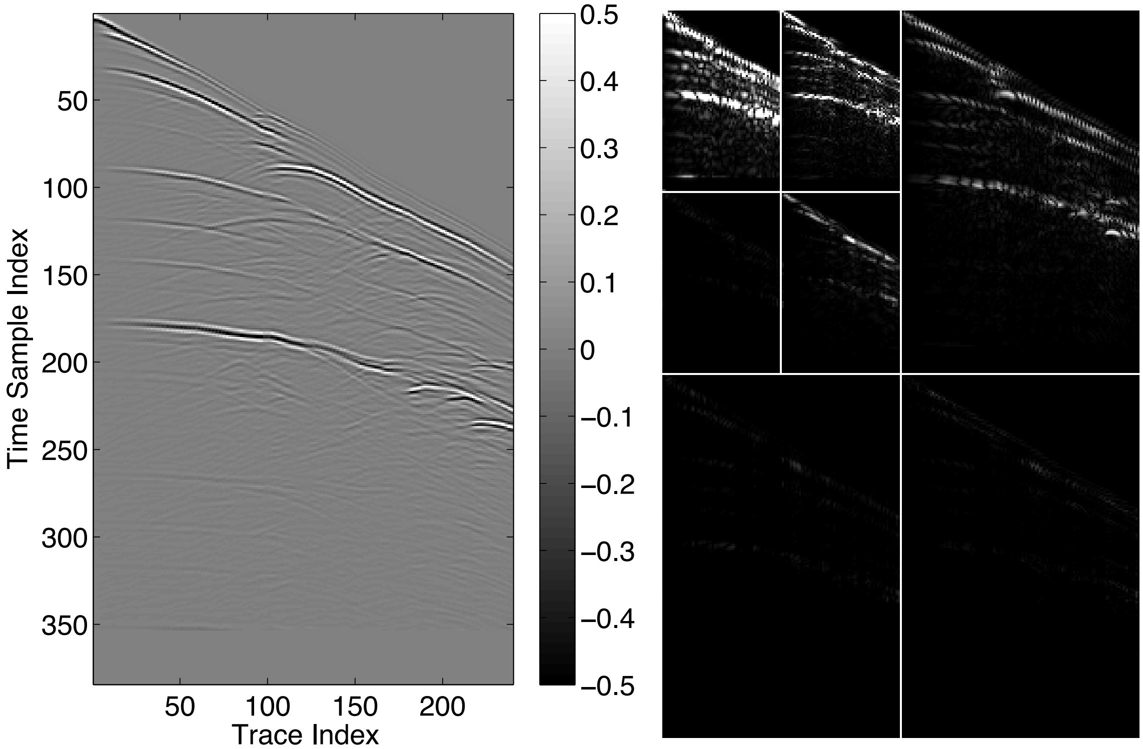
$$\left\{\hat{\mathbf{s}}, \hat{\mathbf{A}}, \mathbf{x}_{ij}\right\} = \underset{\mathbf{s},\mathbf{A},\mathbf{x}}{\operatorname{argmin}} \sum_{(i,j)} \|\boldsymbol{\mathcal{R}}_{ij}\mathbf{s} - \mathbf{\Phi A x}\|_2^2 + \sum_{(i,j)} \mu_{ij}\|\mathbf{x}\|_1 + \lambda\|\mathbf{s} - \mathbf{w}\|_2^2. \tag{2.29}$$

As the previously constructed algorithms suggest, (2.29) can be solved by an iterative three-step algorithm in which each step fixes two unknown estimators and solves the remaining one. The first step assumes a fixed $\hat{\mathbf{s}}$ and $\hat{\mathbf{A}}$ to compute the sparse coefficients $\mathbf{x}_{ij}$ by solving distinct LASSO problems like (2.11) for all $\boldsymbol{\mathcal{R}}_{ij}\hat{\mathbf{s}}$ with a pursuit algorithm. Once this step is done, the fixed $\hat{\mathbf{s}}$ and $\mathbf{x}_{ij}$ are used to update each column of $\mathbf{A}$ by solving the problem like (2.13). Then $\mathbf{s}$ can be estimated by using (2.28). However, the denoised seismic dataset $\hat{\mathbf{s}}$ reduces the noise variance σ^2 which has been considered as known in the preceding two steps. Therefore, before finding the denoised result $\hat{\mathbf{s}}$, a practical implementation would need to perform several more iterations of the sparse representation and dictionary update with the same σ^2. Algorithm 2.4 describes the overall seismic dataset denoising in detail.

2.4.1 Multi-scale Dictionary Learning & Denoising

If the base dictionary $\mathbf{\Phi}$ corresponds to some multi-scale synthesis operator such as the inverse wavelet transform, the optimization problem (2.10) can be modified into the following equivalent form

$$\begin{aligned} \left\{\hat{\mathbf{A}}, \hat{\mathbf{X}}\right\} &= \underset{\mathbf{A},\mathbf{X}}{\operatorname{argmin}} \left\|\mathbf{\Phi}^{\dagger}\mathbf{Y} - \mathbf{AX}\right\|_F^2 \\ &\text{subject to} \quad \begin{cases} \forall i: \ \|\mathbf{x}_i\|_1 \le t \\ \forall j: \ \|\mathbf{a}_j\|_1 \le p, \|\mathbf{\Phi a}_j\|_2 = 1. \end{cases} \end{aligned} \tag{2.30}$$



(a) Synthetic seismic dataset

(b) Wavelet subbands

Figure 2.2: Synthetic seismic dataset and its wavelet subbands

Assuming that $\mathbf{\Phi}$ corresponds to an orthogonal transform, or equivalently, the synthesis operator of the transform, then $\mathbf{\Phi}^{\dagger}$ denotes the analysis operator of the transform. The optimization problem in (2.30) suggests that the sparse matrix $\mathbf{A}$ can be learned not only in the raw data domain, but also in the analysis domain of a transform in which the seismic data is decomposed into multi-scale subbands. Since multi-scale transforms can capture the directional details of seismic wave fronts in different subbands, coefficients tend to be highly correlated across directions and scales. It is essential to learn this structure similarity through some adaptive dictionaries. Figure 2.2 shows $B = 7$ wavelet subbands of a synthetic seismic dataset after a two-scale decomposition. Different subbands are separated by white lines. Therefore, in the multi-scale dictionary learning process, each subband can be treated individually. Separate sparse sub-dictionaries are trained for each subband first, and are then applied to denoise the subband coefficients using the patch-based approach. As Figure

2.2(b) shows, the size of the subband is smaller in the deeper decomposition scale. This enables the patch-based approach to have a global perspective since even a small patch in the deeper scale represents a larger area in the data domain. Algorithm 2.5 presents the complete process of multi-scale dictionary learning.

```
Input: Seismic data $\mathbf{y}$, multi-scale transform that can generate $B$ subbands, signal
       sparsity level $t^{(b)}$, $b = 1, \ldots, B$ and number of iterations $k$
Output: Sparse sub-dictionary $\mathbf{A}^{(b)}$, sparse representation matrix $\mathbf{X}^{(b)}$ and
        approximation error $e^{(b)} \leftarrow \left\|\mathbf{Z}^{(b)} - \mathbf{A}^{(b)}\mathbf{X}^{(b)}\right\|_F^2$, $b = 1, \ldots, B$
Initialization : $\forall b = 1, \ldots B :\ \mathbf{A}^{(b)} \leftarrow \mathbf{A}_0$; $\mathbf{z}^{(b)} \leftarrow (\mathbf{\Phi}^{\dagger}\mathbf{y})^{(b)}$;
for $b \leftarrow 1$ to $B$ do
    Extract Patches: extract overlapping patches from the band coefficients $\mathbf{z}^{(b)}$ to
    construct the training set matrix $\mathbf{Z}^{(b)}$;
    Dictionary Learning: learn the subband-related sparse sub-dictionary $\mathbf{A}^{(b)}$
    using the K-SVD algorithm (Algorithm 2.1) with input $\left\{\mathbf{Z}^{(b)}, t^{(b)}, K\right\}$
end
```

Algorithm 2.5: Multi-scale sparse K-SVD algorithm

As before, let $\mathbf{y}$ denote the vectorized seismic dataset contaminated by noise, then its multi-scale transform result is a collection of coefficient subbands $\mathbf{z}^{(b)} = (\mathbf{\Phi}^{\dagger}\mathbf{y})^{(b)}$ where b is the subband index. For the multi-scale wavelet transform with S decomposition scales, $b = 1, \ldots, B = 3S + 1$. After breaking up each subband into patches and grouping them together in the columns of training sets $\mathbf{Z}^{(b)}$, the multi-scale dictionary learning problem can be expressed as

$$\begin{aligned}\forall b :\ \left\{\hat{\mathbf{A}}^{(b)}, \hat{\mathbf{X}}^{(b)}\right\} &= \underset{\mathbf{A}^{(b)}, \mathbf{X}^{(b)}}{\operatorname{argmin}} \left\|\mathbf{Z}^{(b)} - \mathbf{A}^{(b)}\mathbf{X}^{(b)}\right\|_F^2 \\ \text{subject to} &\quad \begin{cases} \forall i :\ \left\|\mathbf{x}_i^{(b)}\right\|_1 \leq t \\ \forall j :\ \left\|\mathbf{a}_j^{(b)}\right\|_2 = 1. \end{cases}\end{aligned} \tag{2.31}$$

Similar to the global denoising problem (2.25), the sparse coding of $\mathbf{z}^{(b)}$ over the sub-dictionary $\mathbf{A}^{(b)}$ as well as the denoising process of $\mathbf{z}^{(b)}$ can be formulated as follows

$$\begin{aligned}\forall b :\ \left\{\mathbf{x}_{ij}^{(b)}, \hat{\mathbf{u}}^{(b)}\right\} = \underset{\mathbf{x}, \mathbf{u}^{(b)}}{\operatorname{argmin}} \sum_{ij} \left\|\boldsymbol{\mathcal{R}}_{ij}^{(b)}\mathbf{u}^{(b)} - \mathbf{A}^{(b)}\mathbf{x}\right\|_2^2 + \sum_{ij} \mu_{ij}\|\mathbf{x}\|_1 \\ +\lambda \left\|\mathbf{u}^{(b)} - \mathbf{z}^{(b)}\right\|_2^2\end{aligned} \tag{2.32}$$

where $\hat{\mathbf{u}}^{(b)}$ is the denoised version of $\mathbf{z}^{(b)}$ and its closed-form solution is

$$\hat{\mathbf{u}}^{(b)} = \left(\lambda \mathbf{I} + \sum_{ij} \left[\boldsymbol{\mathcal{R}}_{ij}^{(b)}\right]^T \boldsymbol{\mathcal{R}}_{ij}^{(b)}\right)^{-1} \left(\lambda \mathbf{z}^{(b)} + \sum_{ij} \left[\boldsymbol{\mathcal{R}}_{ij}^{(b)}\right]^T \mathbf{A}^{(b)} \mathbf{x}_{ij}^{(b)}\right). \tag{2.33}$$

Finally, the denoised seismic data $\hat{\mathbf{s}}$ can be obtained by applying the inverse multi-scale transform after all subbands across different scales have been denoised

$$\hat{\mathbf{s}} = \boldsymbol{\Phi}\left(\bigcup_b \hat{\mathbf{u}}^{(b)}\right). \tag{2.34}$$

2.5 Extension to Nonhomogeneous Noise

The double-sparsity dictionary learning method can be extended to the case where the noise is nonhomogeneous. Specifically, the nonhomogeneous noise here refers to the missing traces in addition to the usual additive noise. This problem is very important because real land seismic data acquisition may encounter different sorts of obstacles, such as buildings, highways, fences, etc. These obstacles, coupled with limited recording capacity or budget constraints, result in inadequate or irregular spatial traces in the acquired seismic dataset. Both homogeneous and nonhomogeneous types of noise can produce artifacts in seismic imaging results. Therefore, inpainting (trace interpolation), along with denoising, has attracted much attention in research and has become one essential step in industrial seismic data preprocessing workflow.

Previously, a variety of methods have been developed for seismic dataset inpainting. At the very beginning, [110] proposed a trace interpolation method by wave-equation methods based on the principles of wave physics. Later, methods based on the Fourier transform [43, 77, 146] have been adopted to reconstruct irregularly sampled seismic signals for industrial applications. In the recent decade, multi-scale transform methods are also widely used to fill the gaps among traces based on the sparsity of seismic wave fronts in the transform domain, such as [49, 54, 55, 58, 89, 136, 143]. These methods process the dataset as a whole.

In this section, the patch-based strategy is continually used for seismic dataset inpainting. The sparsity-constrained minimization problem (2.29) is still helpful here, yet cannot be directly solved for inpainting. It would treat all data samples as useful information and try to estimate them with sparse coefficients, including the missing trace samples with invalid values. In order to perform inpainting correctly, only the information from available traces should be considered for dictionary learning. This can be done by introducing a mask vector $\boldsymbol{\beta} \in \{0,1\}^{N_z N_x}$ whose elements are determined by

$$\beta_i = \begin{cases} 1, & w_i \text{ is available} \\ 0, & w_i \text{ is missing.} \end{cases} \tag{2.35}$$

By denoting $\odot$ as the element-wise multiplication between two matrices or two vectors, the following optimization problem, which is a weighted version of (2.29), becomes the key for seismic dataset inpainting

$$\begin{aligned} \left\{\hat{\mathbf{s}}, \hat{\mathbf{A}}, \mathbf{x}_{ij}\right\} = \underset{\mathbf{s},\mathbf{A},\mathbf{x}}{\operatorname{argmin}} \sum_{(i,j)} \|(\boldsymbol{\mathcal{R}}_{ij}\boldsymbol{\beta}) \odot (\boldsymbol{\mathcal{R}}_{ij}\mathbf{s} - \boldsymbol{\Phi}\mathbf{A}\mathbf{x})\|_2^2 + \sum_{(i,j)} \mu_{ij}\|\mathbf{x}\|_1 \\ +\lambda\|\boldsymbol{\beta} \odot (\mathbf{s} - \mathbf{w})\|_2^2. \end{aligned} \tag{2.36}$$

After initializing $\hat{\mathbf{s}} = \mathbf{w}$ and using a fixed $\mathbf{A}$, the sparse representation BPDN problem for each patch $\boldsymbol{\mathcal{R}}_{ij}\hat{\mathbf{s}}$ becomes

$$\begin{aligned} \mathbf{x}_{ij} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \\ \text{subject to} \quad \|(\boldsymbol{\mathcal{R}}_{ij}\boldsymbol{\beta}) \odot (\boldsymbol{\mathcal{R}}_{ij}\hat{\mathbf{s}} - \boldsymbol{\Phi}\mathbf{A}\mathbf{x})\|_2^2 \le \|\boldsymbol{\mathcal{R}}_{ij}\boldsymbol{\beta}\|_0 \cdot \sigma^2, \quad \forall (i,j), \end{aligned} \tag{2.37}$$

where the use of $\boldsymbol{\beta}$ guarantees that the missing traces are not taken into account.

Then, in the process of updating each column $\mathbf{a}_k$ of the matrix $\mathbf{A}$ using the fixed $\hat{\mathbf{s}}$ and calculated $\mathbf{x}_{ij}$, the following problem, which replaces (2.13), needs to be solved:

$$\left\{\mathbf{a}_k, \mathbf{X}_{k,\mathcal{I}_k}^T\right\} = \underset{\mathbf{a},\mathbf{x}}{\operatorname{argmin}} \|\mathbf{B}_k \odot (\mathbf{E}_k - \boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T)\|_F^2 \quad \text{subject to} \quad \begin{cases} \|\mathbf{a}\|_1 \le p \\ \|\boldsymbol{\Phi}\mathbf{a}\|_2 = 1, \end{cases} \tag{2.38}$$

where the matrix $\mathbf{B}_k$ collects $\boldsymbol{\mathcal{R}}_{ij}\boldsymbol{\beta}$ in columns for those (i,j) that satisfy $((i-1)(N_x - n_x+1)+j) \in \mathcal{I}_k$ and it has the same size with $\mathbf{E}_k$. Different from (2.13), this problem is a weighted low-rank approximation problem. Unfortunately, due to the introduction of the element-wise mask matrix $\mathbf{B}_k$, Theorem 2.1 no longer holds for the objective function in (2.38).

Alternatively, [91] put forward a simple but effective iterative algorithm to approach the local minima of the objective function in (2.38). The algorithm is based on the expectation-maximization (EM) procedure in which the expectation step fills in the current estimate of $\boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T$ for all missing elements in $\mathbf{B}_k \odot \mathbf{E}_k$ and the maximization step updates $\boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T$ from the filled-in version of $\mathbf{B}_k \odot \mathbf{E}_k$.

```
Input: $\mathbf{E}_k \in \mathbb{R}^{N\times|\mathcal{I}_k|}$, base dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{N\times L}$, mask matrix $\mathbf{B}_k \in \mathbb{R}^{N\times|\mathcal{I}_k|}$,
       number of iterations $K$
Output: $\mathbf{a}_k \in \mathbb{R}^L$, $\mathbf{X}^T_{k,\mathcal{I}_k} \in \mathbb{R}^{|\mathcal{I}_k|}$
Initialization: $\mathbf{a}_{\text{new}} \leftarrow \mathbf{0}$, $\mathbf{x}_{\text{new}} \leftarrow \mathbf{X}^T_{k,\mathcal{I}_k}$
repeat
    $\mathbf{a}_{\text{old}} \leftarrow \mathbf{a}_{\text{new}}$;
    $\mathbf{x}_{\text{old}} \leftarrow \mathbf{x}_{\text{new}}$;
    Solve the following problem with the assistance of Theorem 2.1
    $\{\mathbf{a}_{\text{new}}, \mathbf{x}_{\text{new}}\} = \begin{cases} \underset{\mathbf{a},\mathbf{x}}{\operatorname{argmin}} \left\| \overbrace{\left[\mathbf{B}_k \odot \mathbf{E}_k + (\mathbf{1}-\mathbf{B}_k)\odot(\boldsymbol{\Phi}\mathbf{a}_{\text{old}}\mathbf{x}^T_{\text{old}})\right]}^{\mathbf{E}'_k} - \boldsymbol{\Phi}\mathbf{a}\mathbf{x}^T \right\|_F^2 \\ \text{subject to} \quad \|\mathbf{a}\|_1 \le p, \|\boldsymbol{\Phi}\mathbf{a}\|_2 = 1 \end{cases};$
until $K$ iterations;
$\mathbf{a}_k \leftarrow \mathbf{a}_{\text{new}}$;
$\mathbf{X}^T_{k,\mathcal{I}_k} \leftarrow \mathbf{x}_{\text{new}}$;
```

Algorithm 2.6: Weighted low-rank approximation algorithm

To put it concretely, Algorithm 2.6 presents the iterative EM-based algorithm that solves (2.38). Every time $\mathbf{a}$ and $\mathbf{d}$ are estimated, with the names $\mathbf{a}_{\text{old}}$ and $\mathbf{x}_{\text{old}}$, they are used to fill in $\mathbf{B}_k \odot \mathbf{E}_k$ by generating a new observation matrix

$$\mathbf{E}'_k \triangleq \mathbf{B}_k \odot \mathbf{E}_k + (\mathbf{1}-\mathbf{B}_k)\odot(\boldsymbol{\Phi}\mathbf{a}_{\text{old}}\mathbf{x}^T_{\text{old}})$$

in the expectation step. Then, in the maximization step, $\mathbf{a}$ and $\mathbf{d}$ are updated by the filled-in observation matrix $\mathbf{E}'_k$

$$\{\mathbf{a}_{\text{new}}, \mathbf{x}_{\text{new}}\} = \underset{\mathbf{a},\mathbf{x}}{\text{argmin}} \left\|\mathbf{E}'_k - \mathbf{\Phi}\mathbf{a}\mathbf{x}^T\right\|_F^2 \quad \text{subject to} \quad \begin{cases} \|\mathbf{a}\|_1 \leq p \\ \|\mathbf{\Phi}\mathbf{a}\|_2 = 1. \end{cases} \tag{2.39}$$

The problem in the form of (2.39) can be solved with the assistance of Theorem 2.1, where $\mathbf{a}_{\text{new}}$ is a solution of a sparse coding problem like (2.15) and $\mathbf{x}_{\text{new}}$ has the form like (2.17). The EM procedure converges to a local minimum very quickly, within only a few ($K \approx 5$) iterations.

Finally, when $\mathbf{A}$ and all $\mathbf{x}_{ij}$ are obtained, the last remaining problem of (2.36) for the inpainting result $\hat{\mathbf{s}}$ is

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\text{argmin}} \sum_{(i,j)} \|\boldsymbol{\mathcal{R}}_{ij}\mathbf{s} - \mathbf{\Phi}\mathbf{A}\mathbf{x}_{ij}\|_2^2 + \lambda\|\boldsymbol{\beta} \odot (\mathbf{s} - \mathbf{w})\|_2^2. \tag{2.40}$$

Note that the mask $\boldsymbol{\mathcal{R}}_{ij}\boldsymbol{\beta}$ has been removed in front of the reconstruction misfit $\boldsymbol{\mathcal{R}}_{ij}\mathbf{s} - \mathbf{\Phi}\mathbf{A}\mathbf{x}_{ij}$ since right now the entire $\mathbf{s}$ is being reconstructed including the missing traces. Similar to the form of (2.28), the closed-form solution of (2.40) is

$$\hat{\mathbf{s}} = \left(\lambda\text{diag}(\boldsymbol{\beta}) + \sum_{(i,j)} \boldsymbol{\mathcal{R}}_{ij}^{\dagger}\boldsymbol{\mathcal{R}}_{ij}\right)^{-1} \left(\lambda(\boldsymbol{\beta} \odot \mathbf{w}) + \sum_{(i,j)} \boldsymbol{\mathcal{R}}_{ij}^{\dagger}\mathbf{\Phi}\mathbf{A}\mathbf{x}_{ij}\right). \tag{2.41}$$

Algorithm 2.7 describes the overall seismic dataset inpainting in detail.

2.6 Numerical Experiments

In this section, the dictionary learning method is used to attenuate the noise and fill in the missing traces in seismic data. The performance of the proposed method is also compared with other denoising methods using the fixed contourlet and curvelet transforms. The seismic dataset used in the experiments are synthesized 2D pre-stack shot records that are available for download at public domains in Society of Exploration Geophysicists (SEG) and Madagascar [81]. Assuming the seismic noise is caused by a diversity of different, spatially distributed, mostly uncorrelated but low-frequency sources, it can be modeled as zero-mean white additive Gaussian

```
Input: Vectorized noisy seismic dataset $\mathbf{w} \in \mathbb{R}^{N_z N_x}$ with missing traces, mask
        vector $\boldsymbol{\beta} \in \mathbb{R}^{N_z N_x}$ patch height $n_z$, patch width $n_x$, $N = n_z n_x$, base
        dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{N \times L}$, number of training iterations $K_T$, number of atom
        update iterations $K_U$
Output: Inpainted seismic dataset $\hat{\mathbf{s}} \in \mathbb{R}^{N_z N_x}$, sparse matrix $\mathbf{A} \in \mathbb{R}^{L \times L}$, sparse
        coefficient matrix $\mathbf{X} \in \mathbb{R}^{L \times (N_z - n_z + 1)(N_x - n_x + 1)}$
Initialization : $\hat{\mathbf{s}} \leftarrow \mathbf{w}$, $\mathbf{A} \leftarrow \mathbf{I}$, $\mathbf{X} \leftarrow \mathbf{0}$
repeat
    // Sparse Representation Stage
    for $i \leftarrow 1$ to $N_z - n_z + 1$ do
        for $j \leftarrow 1$ to $N_x - n_x + 1$ do
            $\mathbf{x}_{ij} \leftarrow \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1$ s.t. $\|(\boldsymbol{\mathcal{R}}_{ij}\boldsymbol{\beta}) \odot (\boldsymbol{\mathcal{R}}_{ij}\hat{\mathbf{s}} - \boldsymbol{\Phi}\mathbf{A}\mathbf{x})\|_2^2 \leq \|\boldsymbol{\mathcal{R}}_{ij}\boldsymbol{\beta}\|_0 \cdot \sigma^2$;
            Place $\mathbf{x}_{ij}$ into $\mathbf{X}$ as a column with index $(i-1)(N_x - n_x + 1) + j$;
        end
    end
    // Dictionary Update Stage
    for $k \leftarrow 1$ to $L$ do
        $\mathcal{I}_k \leftarrow \{r | 1 \leq r \leq (N_z - n_z + 1)(N_x - n_x + 1), x_{kr} \neq 0\}$;
        $\mathbf{B}_k$ collects $\boldsymbol{\mathcal{R}}_{ij}\boldsymbol{\beta}$ in columns for those $(i,j)$ that satisfy
        $((i-1)(N_x - n_x + 1) + j) \in \mathcal{I}_k$;
        // Atom removal
        $\mathbf{a}_{\text{new}} \leftarrow \mathbf{a}_k \leftarrow \mathbf{0}$;
        $\mathbf{x}_{\text{new}} \leftarrow \mathbf{X}_{k,\mathcal{I}_k}^T$;
        $\mathbf{E}_k \leftarrow \mathbf{Y}_{\mathcal{I}_k} - \boldsymbol{\Phi}\mathbf{A}\mathbf{X}_{\mathcal{I}_k}$;
        // Atom updating
        repeat
            $\mathbf{a}_{\text{old}} \leftarrow \mathbf{a}_{\text{new}}$;
            $\mathbf{x}_{\text{old}} \leftarrow \mathbf{x}_{\text{new}}$;
            $\mathbf{E}_k' \leftarrow \mathbf{B}_k \odot \mathbf{E}_k + (\mathbf{1} - \mathbf{B}_k) \odot (\boldsymbol{\Phi}\mathbf{a}_{\text{old}}\mathbf{x}_{\text{old}}^T)$;
            $\mathbf{x}_{\text{old}} \leftarrow \mathbf{x}_{\text{old}} / \|\mathbf{x}_{\text{old}}\|_2$;
            $\mathbf{a}_{\text{new}} \leftarrow \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{E}_{\mathbf{k}}'\mathbf{x}_{\text{old}} - \boldsymbol{\Phi}\mathbf{a}\|_2^2 \quad$ subject to $\quad \|\mathbf{a}\|_1 \leq p$;
            $\mathbf{a}_{\text{new}} \leftarrow \mathbf{a}_{\text{new}} / \|\boldsymbol{\Phi}\mathbf{a}_{\text{new}}\|_2$;
            $\mathbf{x}_{\text{new}} \leftarrow [\mathbf{E}_k']^T \boldsymbol{\Phi}\mathbf{a}_{\text{new}}$;
        until $K_U$ atom update iterations;
        $\mathbf{a}_k \leftarrow \mathbf{a}_{\text{new}}$;
        $\mathbf{X}_{k,\mathcal{I}_k} \leftarrow \mathbf{x}_{\text{new}}^T$;
    end
    for $j \leftarrow 1$ to $L$ do
        Atom_Replacing($\boldsymbol{\Phi}\mathbf{a}_k$);
    end
until $K_T$ training iterations;
// Inpainting Stage
$\hat{\mathbf{s}} \leftarrow \left(\lambda \operatorname{diag}(\boldsymbol{\beta}) + \sum_{(i,j)} \boldsymbol{\mathcal{R}}_{ij}^\dagger \boldsymbol{\mathcal{R}}_{ij}\right)^{-1} \left(\lambda(\boldsymbol{\beta} \odot \mathbf{w}) + \sum_{(i,j)} \boldsymbol{\mathcal{R}}_{ij}^\dagger \boldsymbol{\Phi}\mathbf{A}\mathbf{x}_{ij}\right)$
```

Algorithm 2.7: Inpaint seismic dataset using the double-sparsity dictionary learned on patches from the noisy dataset with missing traces

noise with standard deviation σ low-pass filtered at a stopband frequency (30 Hz in the experiments). For the sake of numerical stability and better performance, a standard normalization step is applied to rescale each data into the range $[-1, 1]$ after subtracting its mean. Two different base dictionaries $\mathbf{\Phi}$ are used in the experiments to learn the sparse matrix $\mathbf{A}$ and thereafter the overall dictionary $\mathbf{D} = \mathbf{\Phi A}$. One represents the single-scale discrete cosine transform (DCT) and another the multi-scale discrete wavelet transform (DWT).

As one of the most commonly used quality metrics for the comparison of denoising performance, peak signal-to-noise ratio (PSNR) is used in the experiments, which is defined as

$$\mathrm{PSNR} \triangleq 20 \log_{10} \left(\frac{\mathbf{s}_{\max} \sqrt{N_z N_x}}{\|\mathbf{s} - \hat{\mathbf{s}}\|_2} \right) \tag{2.42}$$

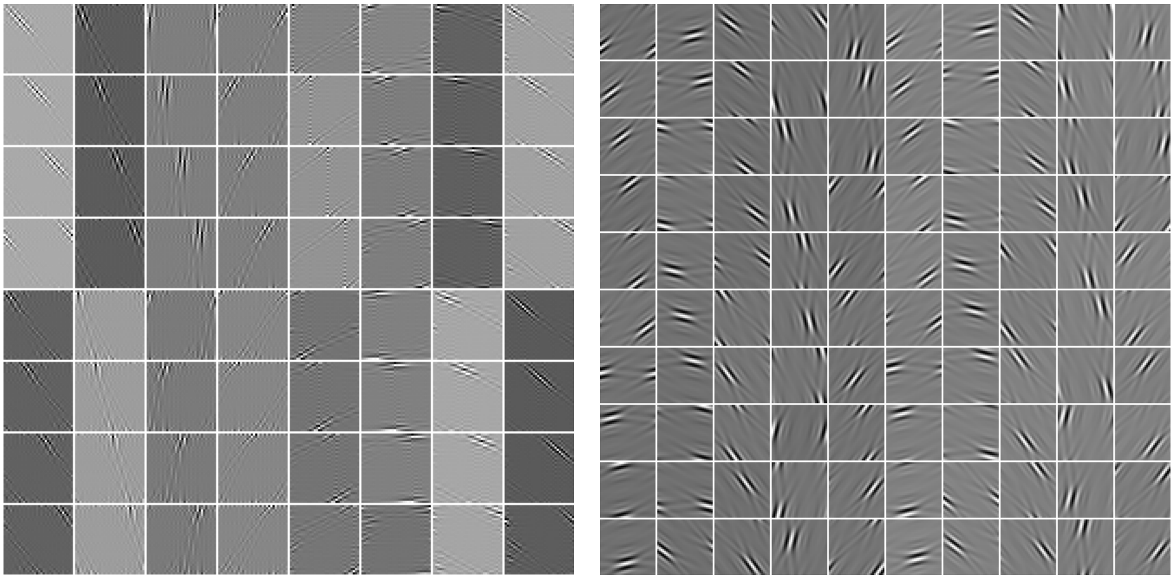
where $\mathbf{s}_{\max}$ is the maximum possible value of the seismic data after normalization, and N_x, N_z are the numbers of traces and time samples per trace, respectively.

2.6.1 Denoising with Fixed Transforms

This subsection investigates the prominent multi-scale directional transforms including the contourlet and curvelet transforms for seismic data denoising. The contourlet transform [34, 35] can capture smooth contours in a seismic dataset based on a Laplacian Pyramid decomposition followed by directional filter banks applied on each bandpass subband. Its atom elements are depicted in Figure 2.3(a). Based on the frequency partition technique, the curvelet transform is able to represent curved singularities more precisely with needle-shaped atom elements, which are shown in Figure 2.3(b). In order to perform a due diligence comparison, the Fast Discrete Curvelet Transform [20, 33], which is the latest curvelet implementation, is used in experiments. Since both transforms are able to represent a seismic dataset with sparse coefficients, the following program formulation can be used for denoising,

$$\begin{cases} \hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{w} - \mathbf{\Phi}\mathbf{x}\|_2^2 \leq N_z N_x \sigma^2 \\ \hat{\mathbf{s}} = \mathbf{\Phi}\hat{\mathbf{x}} \end{cases} \tag{2.43}$$

where $\mathbf{\Phi}$ refers to the dictionary of the contourlet/curvelet synthesis operator.

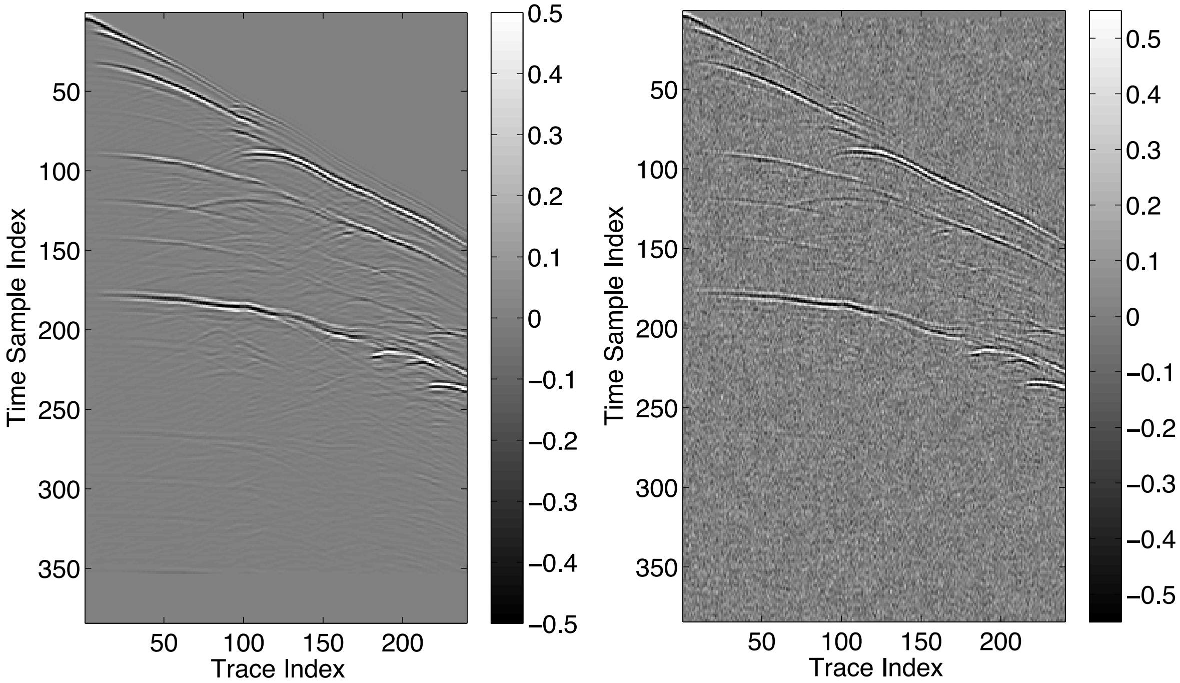


(a) Contourlet Atoms

(b) Curvelet Atoms

Figure 2.3: Atom elements of (a) contourlets and (b) curvelets on different scales and directions in spatial domain

The public seismic dataset in the following experiments is provided by BP [48] as part of the Madagascar software [81]. It has $N_x = 240$ traces and each trace contains $N_z = 384$ time samples. Figure 2.4(a) is the original noise-free seismic dataset for reference. Its noisy version contaminated by (low-pass filtered) Gaussian noise with $\sigma = 0.1$ is shown in Figure 2.4(b), whose PSNR = 22.62 dB. The BPDN results based on the contourlet and curvelet transforms are provided in Figure 2.5(a) and Figure 2.5(c) with PSNR = 29.02 dB and 29.58 dB, respectively, while the error panel figures that show the difference between the reconstructed data and the original data are given in Figure 2.5(b) and Figure 2.5(d), respectively. It is obvious that most of the random noise is attenuated after many small coefficients are suppressed by the



(a) Original seismic dataset

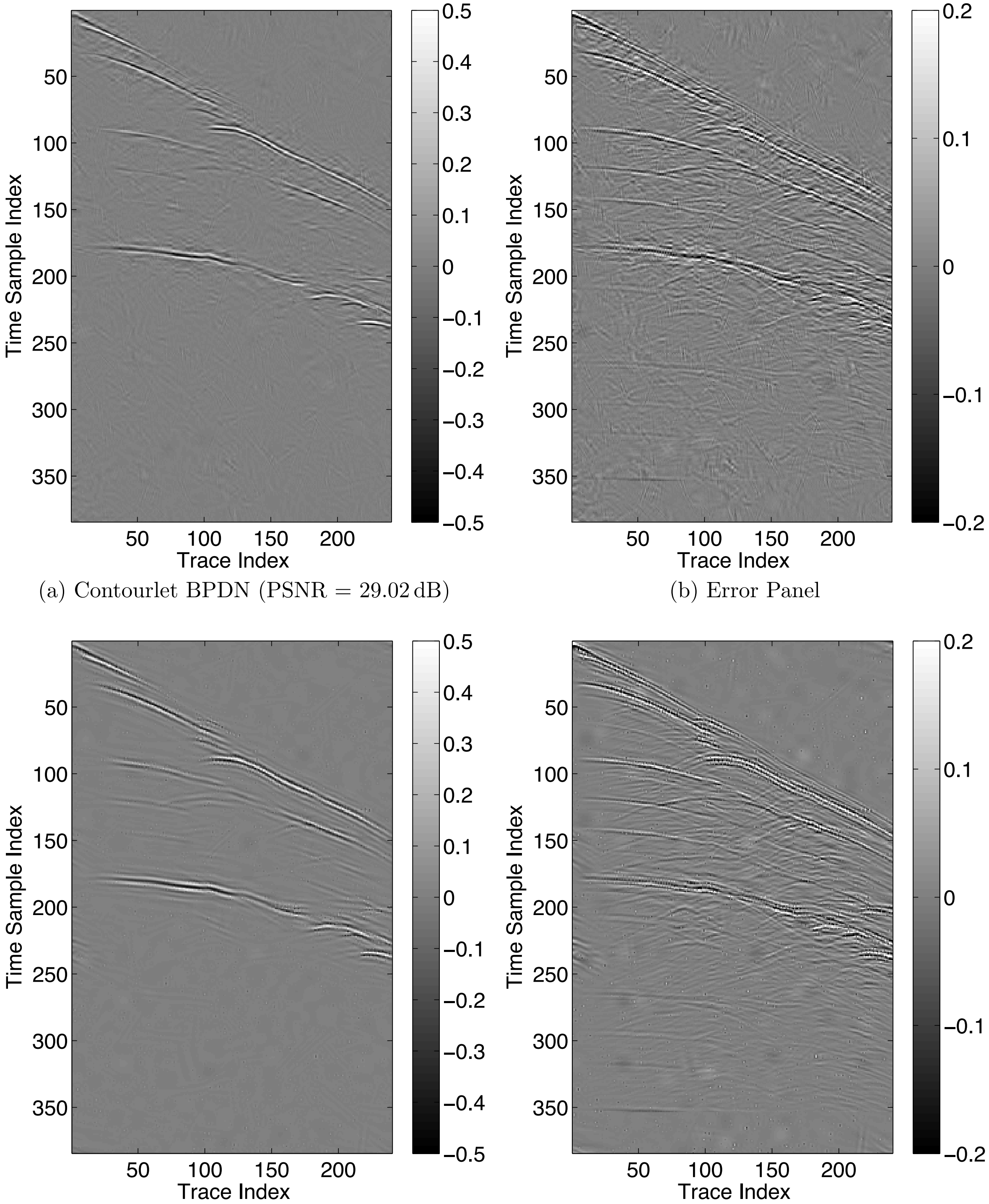
(b) Noisy seismic dataset (PSNR = 22.62 dB)

Figure 2.4: The original and noisy seismic dataset

BPDN algorithm. However, many pseudo-Gibbs artifacts are produced around the wave fronts, especially in the contourlet case where the atom elements have less sharp directional features than curvelets. These artifacts that do not exist in the original dataset may be harmful for further processing such as migration and full waveform tomography.

2.6.2 Denoising with Single-scale Dictionary Learning

The following experiments provide seismic dataset denoising performance results using the double-sparsity dictionary learning method based on the sparse K-SVD algorithm. It starts by dividing the noisy dataset into overlapping patches with size $n_z \times n_x = 16 \times 16$ each, and randomly choosing 10000 among them as a training set for the sparse K-SVD algorithm. A single-scale non-redundant $N \times N = 256 \times 256$ DCT matrix has been selected as the base dictionary for the sparse K-SVD algorithm and the sparse matrix $\mathbf{A}$ is initialized to identity. Therefore, the size of the overall



(a) Contourlet BPDN (PSNR = 29.02 dB)

(b) Error Panel

(c) Curvelet BPDN (PSNR = 29.58 dB)

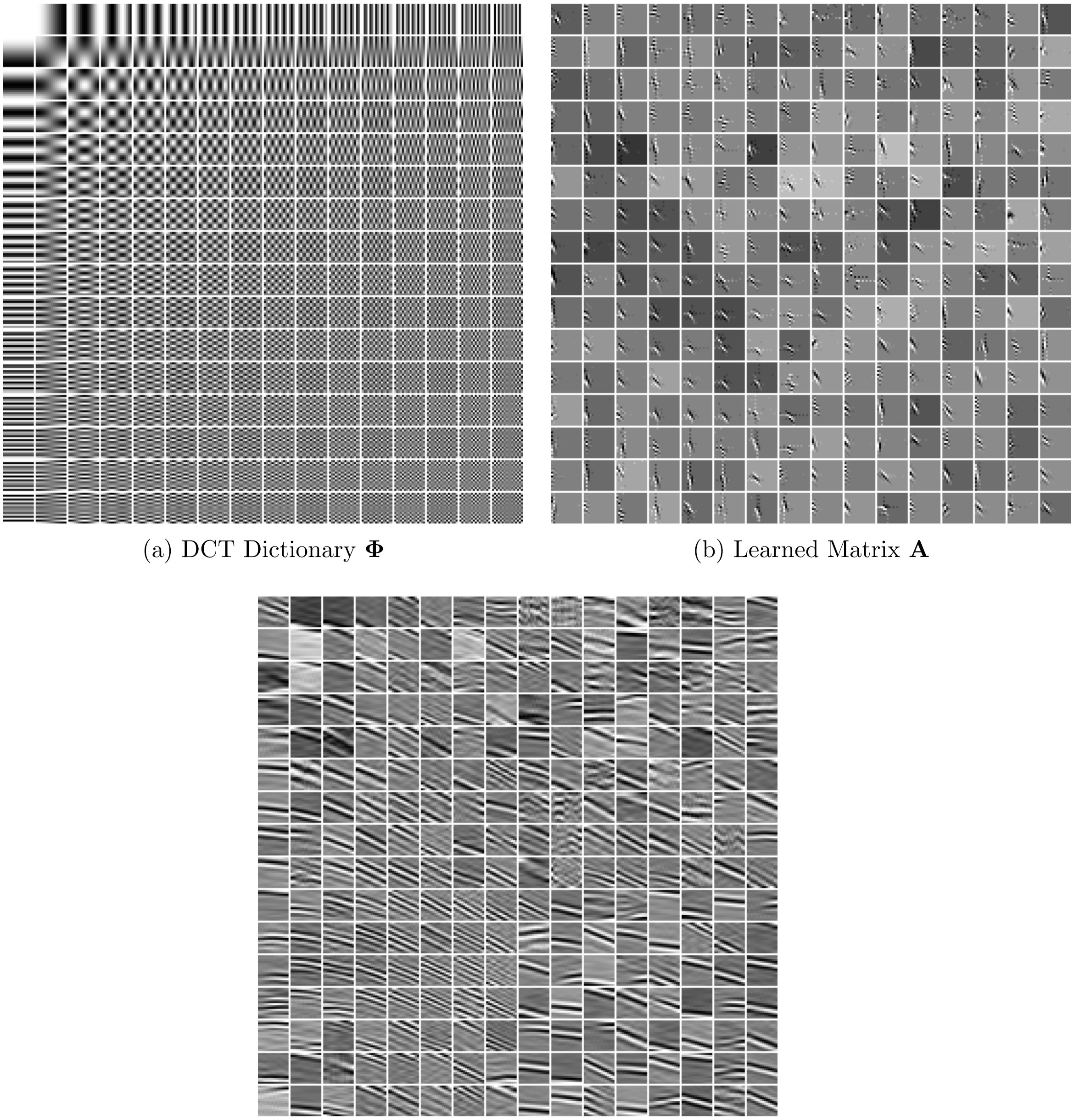
(d) Error Panel

Figure 2.5: Denoised results based on BPDN using the fixed multi-scale transforms without dictionary learning: (a) denoised result by contourlet-based BPDN method (PSNR = 29.02 dB), (b) is the difference of (a) to the original data, (c) denoised result by curvelet-based BPDN method (PSNR = 29.58 dB), and (d) is the difference between (c) and the original data.

learned dictionary $\mathbf{D} = \mathbf{\Phi A}$ is 256×256. The atom sparsity level p is set to 25, implying that the overall dictionary atoms are linear combinations of a small number of arbitrary DCT atoms.

The non-redundant DCT base dictionary is demonstrated in Figure 2.6(a). Figure 2.6(b) depicts the learned sparse matrix $\mathbf{A}$ obtained by running the sparse K-SVD algorithm for $K = 20$ training iterations. Figure 2.6(c) shows the overall learned dictionary $\mathbf{D} = \mathbf{\Phi A}$, in which each atom visualized in a block is a linear combination of all DCT atoms visualized as blocks in Figure 2.6(a), with the coefficients visualized in the block of the same position in Figure 2.6(b). It is obvious that some primary directional features in the seismic wave fronts are characterized by the dictionary atoms. These atoms are more adaptive to the dataset when compared to the fixed directional transforms such as the contourlet, or curvelet, which exhibits atoms in all directions. Therefore, the denoising result shown in Figure 2.7(a) is improved to PSNR = 32.00 dB and the corresponding error panel is shown in Figure 2.7(b). Particularly, since all atoms in the learned dictionary are useful and well representative for sparse coding and patch denoising, the problem of pseudo-Gibbs artifacts is solved. Zoom-in denoising results are demonstrated in Figure 2.8. It is worth noting that the result of BPDN with the curvelet transform in Figure 2.8(a) introduces pseudo-Gibbs artifacts which cannot be ignored, while the dictionary learning method based on the sparse K-SVD algorithm solves this problem, as shown in Figure 2.8(b).

Figure 2.9(a) compares the performance of the dictionary learning based denoising method to the curvelet BPDN method versus different noise levels σ where 20000 training patches are used to learn the sparse matrix $\mathbf{A}$ with atom sparsity levels $p = 25$ and $p = 100$. This result indicates a significant improvement by using an adaptive dictionary based on a fundamental transform. The denoising method performs quite consistently for different settings, as can be seen from the mean and error bars in Figure 2.9(b) and Figure 2.9(c). Figure 2.10 compares PSNR results of denoised

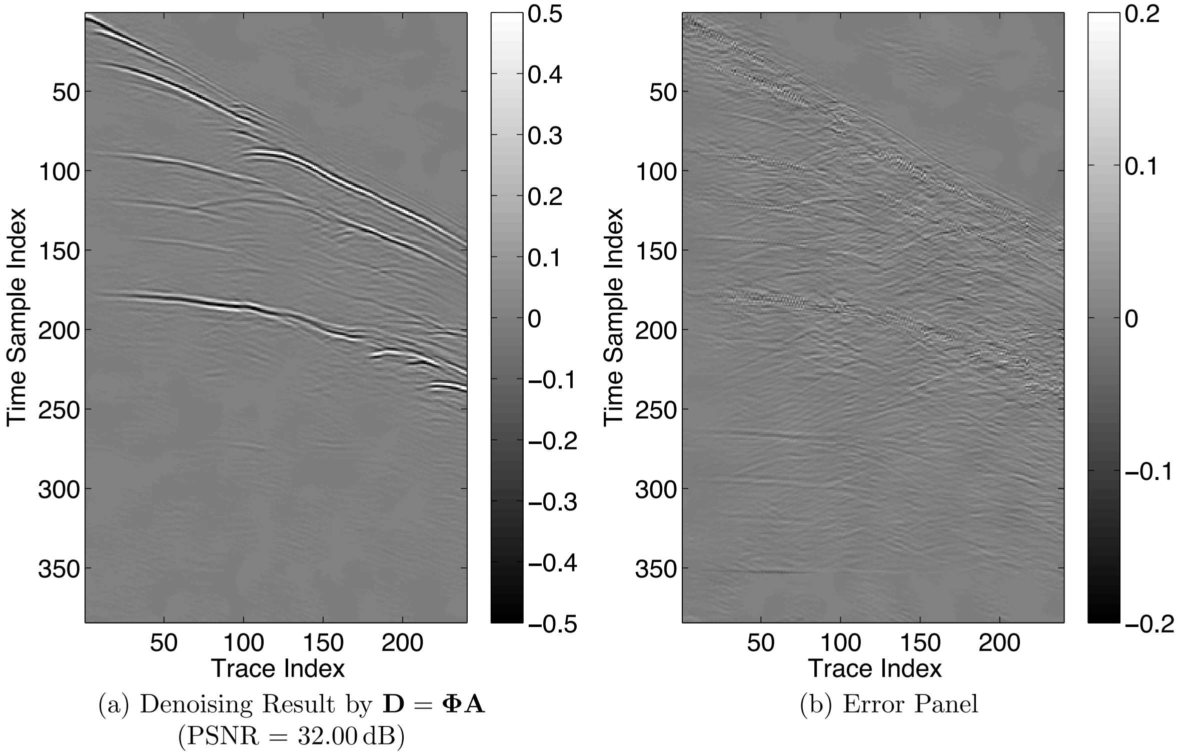


(a) DCT Dictionary $\mathbf{\Phi}$

(b) Learned Matrix $\mathbf{A}$

(c) Overall Learned Dictionary $\mathbf{D} = \mathbf{\Phi}\mathbf{A}$

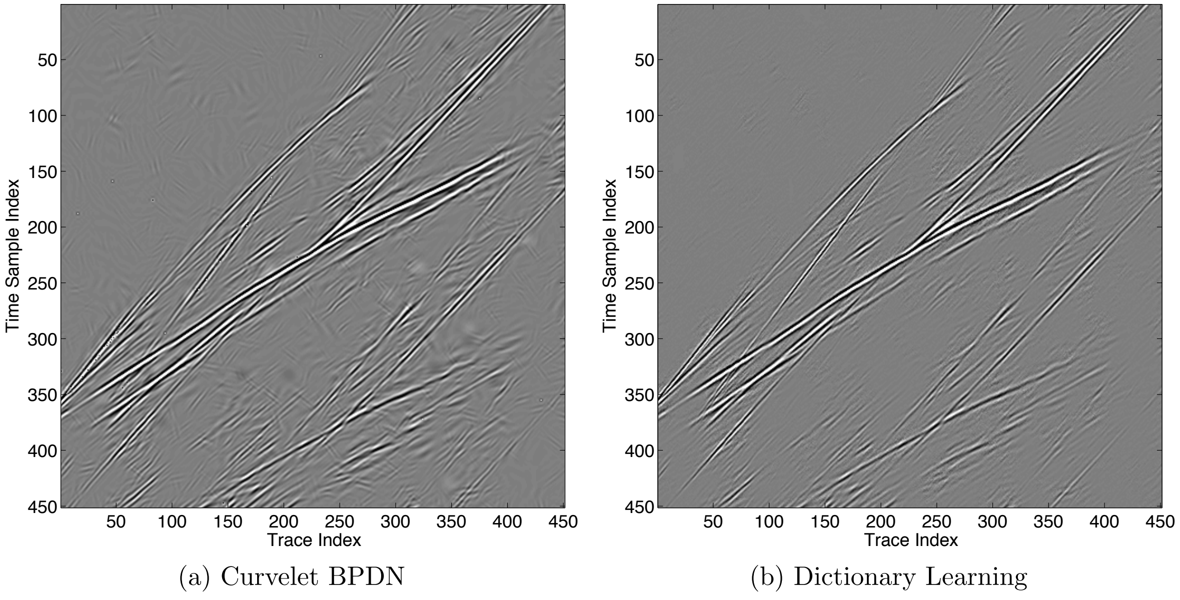
Figure 2.6: Base dictionary (DCT) and learned dictionaries



(a) Denoising Result by $\mathbf{D} = \mathbf{\Phi A}$ (PSNR = 32.00 dB)

(b) Error Panel

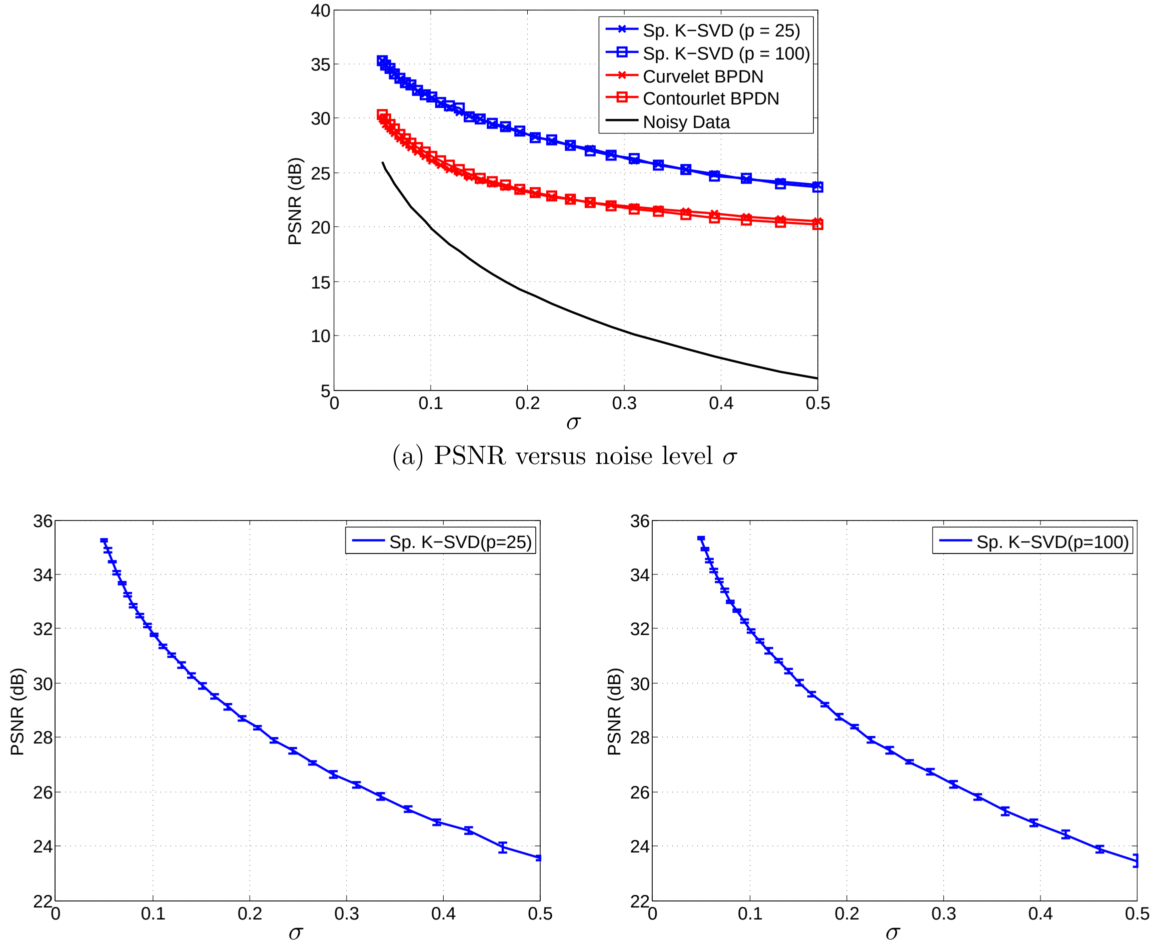
Figure 2.7: Denoised result of dictionary learning method using DCT matrix as the base dictionary is shown in (a) and (b) is the difference of (a) to the original data.



(a) Curvelet BPDN

(b) Dictionary Learning

Figure 2.8: Pseudo-Gibbs artifacts

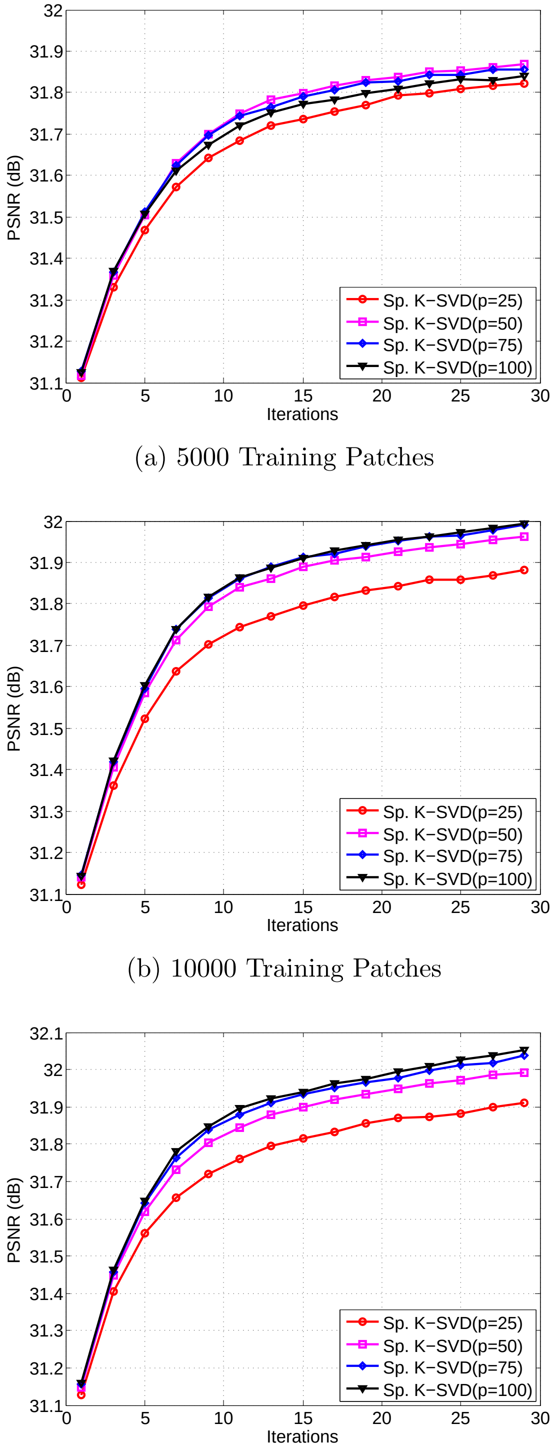


(a) PSNR versus noise level σ

(b) Mean and error bar ($p = 25$)

(c) Mean and error bar ($p = 100$)

Figure 2.9: PSNR versus noise level σ with 20000 training patches for dictionary learning



(a) 5000 Training Patches

(b) 10000 Training Patches

(c) 20000 Training Patches

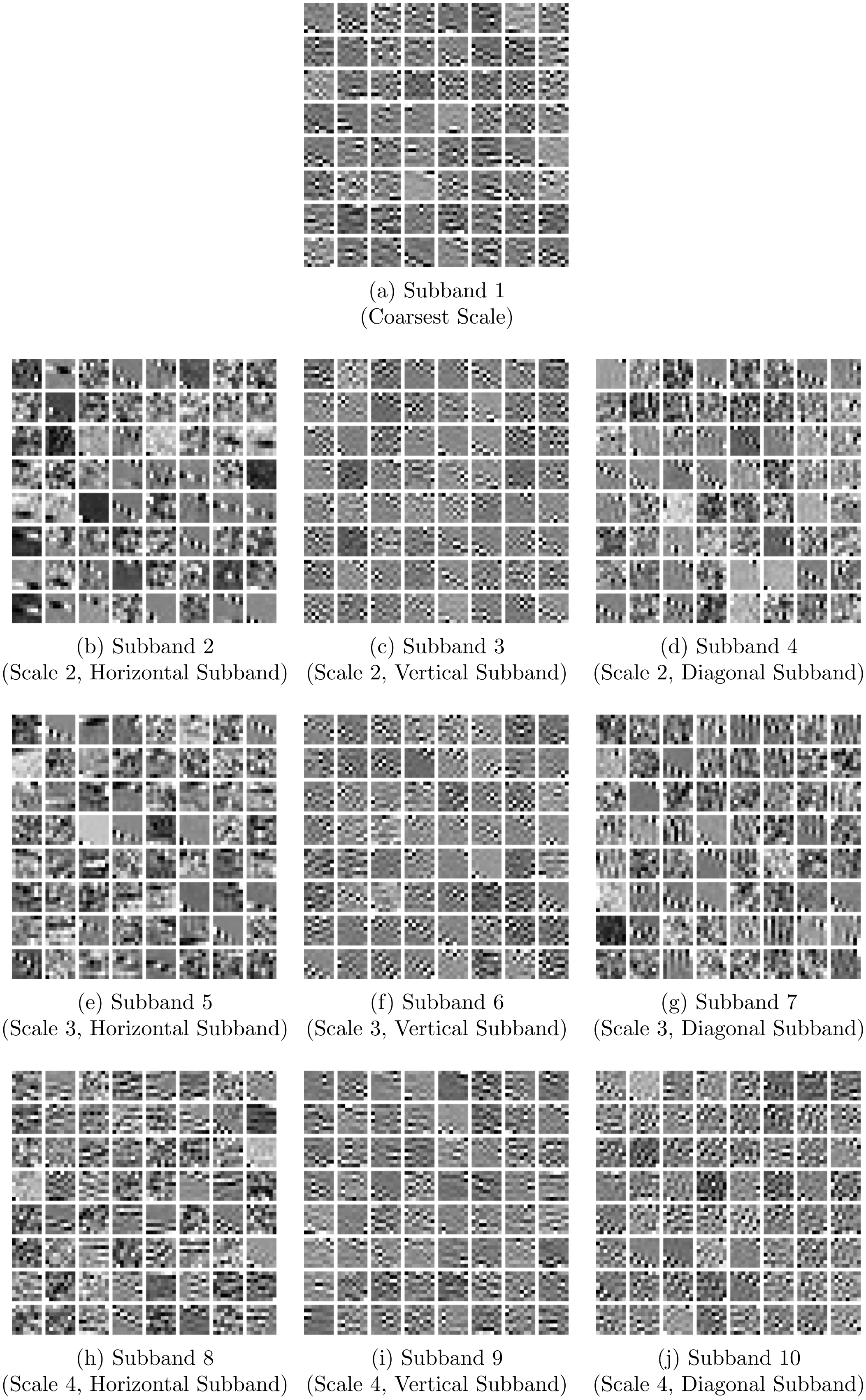
Figure 2.10: PSNR versus training iterations K with different number of training patches, and sparsity levels p

seismic data during training iterations of the sparse K-SVD algorithm under different parameter settings. 5000, 10000 and 20000 training patches are randomly selected from the noisy dataset to learn dictionaries, with each dictionary parameterized with different atom sparsity levels p ranging from 25 to 100. As the number of training iterations or training patches of the algorithm increases, it can be observed that the denoising performance consistently improves. These performance curves motivate a way to choose parameters heuristically.

2.6.3 Denoising with Multi-scale Dictionary Learning

In this experiment, the multi-scale sparse K-SVD algorithm (Algorithm 2.5) is used to learn separate and sparse sub-dictionaries for the sparse coding and denoising of transform coefficients in different subbands. The final denoised seismic data is obtained by applying the inverse transform on the denoised transform coefficients.

The Daubechies 8-tap wavelet transform [32], whose analysis operator is $\mathbf{\Phi}^{\dagger}$, is used to decompose the seismic dataset with $S = 3$ scales, producing $B = 3S + 1 = 10$ wavelet subbands. The patch size is fixed to $n_z \times n_x = 8 \times 8$ for all 10 subbands, producing 10 dictionaries $\mathbf{A}^{(b)}$ of size $N \times N = 64 \times 64$ for $b = 1, \ldots, 10$. Figure 2.11 visualizes these 10 dictionaries, which are obtained using a total number of 10000 training patches across all subbands after $K = 20$ training iterations. The effective dictionaries $\mathbf{D}^{(b)} = \mathbf{\Phi}\mathbf{A}^{(b)}$ inherit the benefits of multi-scale capabilities of the wavelet transform, while enjoying the adaptability of learning in the transform analysis domain. In order to present the visualization of a single effective atom, one can first generate a coefficient vector of length N with only one nonzero element, multiply such a coefficient vector by the corresponding learned dictionary $\mathbf{A}^{(b)}$, then put the result in the b-th wavelet subband at a specific scale, and finally perform the inverse wavelet transform with the wavelet synthesis operator $\mathbf{\Phi}$. Figure 2.12



(a) Subband 1
(Coarsest Scale)

(b) Subband 2
(Scale 2, Horizontal Subband)

(c) Subband 3
(Scale 2, Vertical Subband)

(d) Subband 4
(Scale 2, Diagonal Subband)

(e) Subband 5
(Scale 3, Horizontal Subband)

(f) Subband 6
(Scale 3, Vertical Subband)

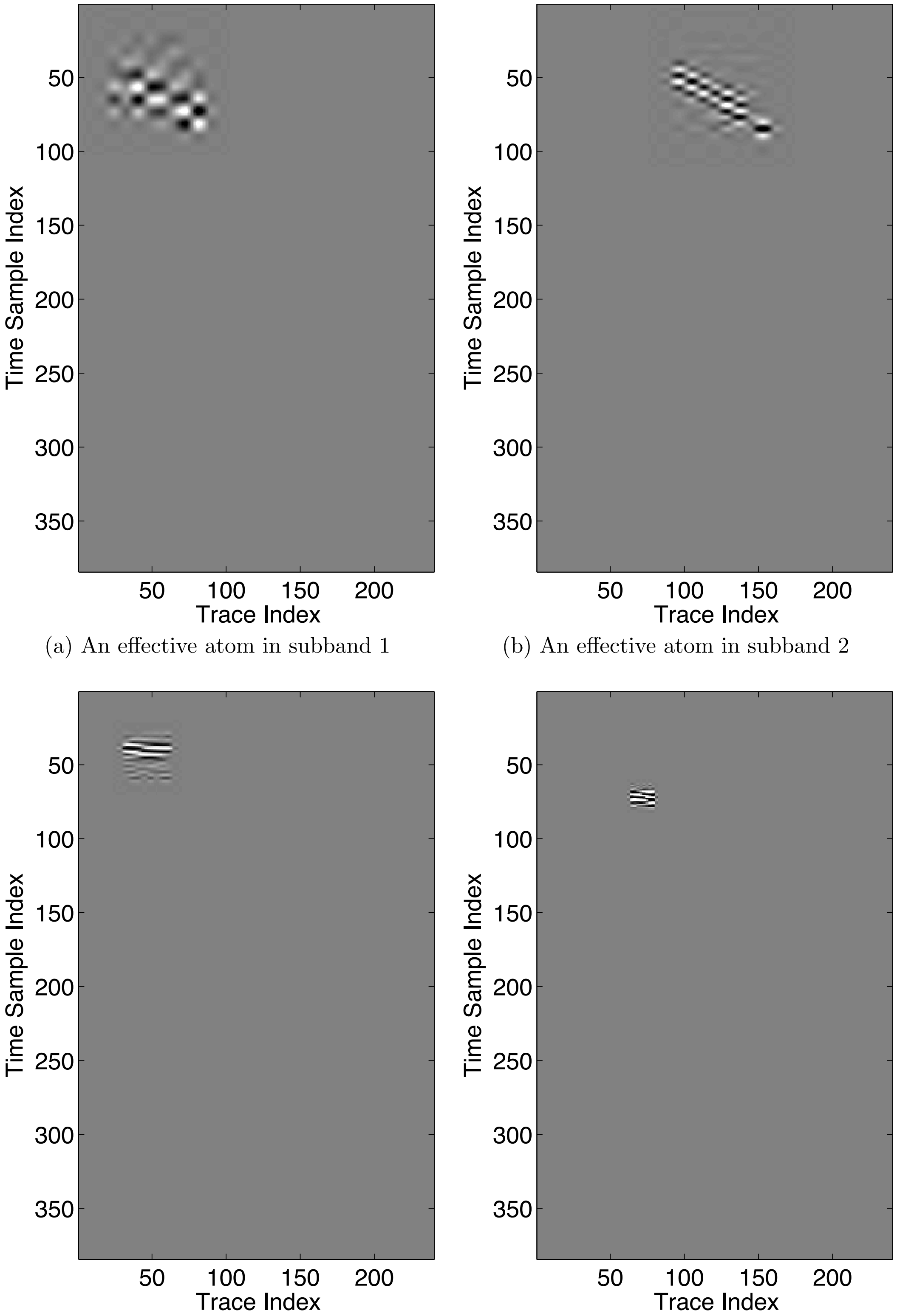
(g) Subband 7
(Scale 3, Diagonal Subband)

(h) Subband 8
(Scale 4, Horizontal Subband)

(i) Subband 9
(Scale 4, Vertical Subband)

(j) Subband 10
(Scale 4, Diagonal Subband)

Figure 2.11: Dictionaries learned from 10 wavelet subbands



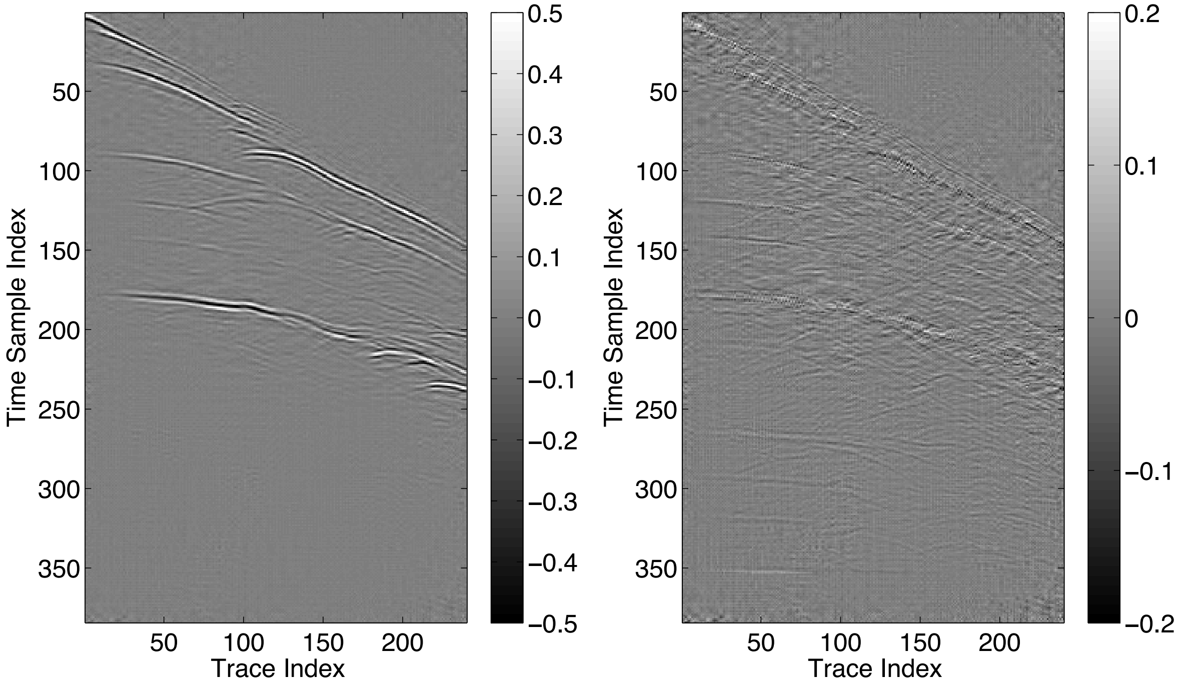
(a) An effective atom in subband 1

(b) An effective atom in subband 2

(c) An effective atom in subband 5

(d) An effective atom in subband 8

Figure 2.12: Visualization of some effective atoms from different scales and subbands trained on the synthesized seismic dataset after 3-scale wavelet decomposition



(a) Denoising Result by $\mathbf{D} = \mathbf{\Phi A}$ (PSNR = 33.01 dB)

(b) Error Panel

Figure 2.13: Denoised result of dictionary learning method using DWT matrix as the base dictionary is shown in (a) and (b) is the difference of (a) to the original data.

visualizes some effective atoms from different scales and subbands. Due to the multi-scale property of the wavelet transform, patches of the same size in different scales correspond to different areas in the original dataset domain. For this 3-scale wavelet decomposition, the size of subbands $b = 1, 2, 3, 4$ are only 1/64-th of the original dataset size, thus training 8×8 patches in these subbands yields effective atoms of size 64×64, as shown in Figures 2.12(a) and 2.12(b). The size of subbands $b = 5, 6, 7$ are 1/16-th of the original dataset size, so that 8×8 patches in these subbands are trained into 32×32 effective atoms, as shown in Figure 2.12(c). Similarly, as shown in Figure 2.12(d), training 8×8 patches in the subbands $b = 8, 9, 10$, whose size are 1/4-th of the original dataset size, produces 16×16 effective atoms. Therefore, it can be verified that these atoms are multi-scale, localized and adapted to subbands.

When the noise level $\sigma = 0.1$, the denoising result in Figure 2.13(a) and the

Table 2.1: PSNR comparison in decibels of the denoised seismic dataset between single-scale and multi-scale dictionary learning

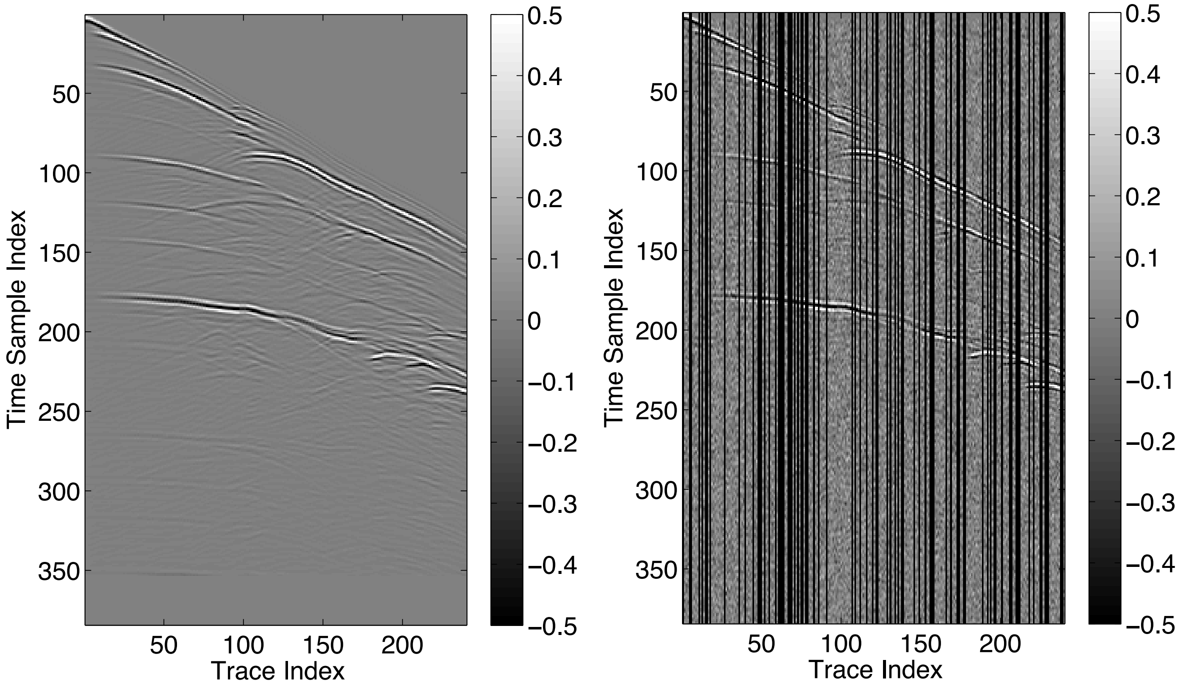
σ	0.05	0.1	0.15	0.2	0.25	0.3
Single-scale	35.26	32.00	29.89	28.36	27.38	26.40
Multi-scale	36.57	33.01	30.73	29.07	27.62	26.51

corresponding error panel in Figure 2.13(b) show that this scheme outperforms the single-scale method by about 1 dB under a similar combination of parameters. Comparing to the results shown in Figure 2.8, this result achieves better performance by using many fewer training patches. Table 2.1 compares the PSNR performance of the same denoised seismic dataset between the single-scale and multi-scale dictionary learning approaches. Choosing a multi-scale base dictionary such as wavelets allows the dictionary learning to work in each subband and squeezes more sparsity out of the signals that have already been sparsified. The seismic dataset denoising results have benefited from these properties.

2.6.4 Inpainting with Dictionary Learning

The following experiments provide seismic dataset inpainting performance results based on the double-sparsity dictionary learning method with nonhomogeneous noise extension. Figure 2.14 shows the original and noisy seismic datasets provided by BP [48, 81] with 33% missing traces whose indices are randomly selected between 1 and $N_x = 240$ and each trace has $N_z = 384$ time samples. Note that all the missing traces have Not-a-Number (NaN) values and their corresponding values in the mask vector $\boldsymbol{\beta}$ are set to zeros. Besides the NaN-valued missing traces, white Gaussian noise with $\sigma = 0.1$ is also added to contaminate the available traces.

First, as baseline experiments, the fixed multi-scale contourlet and curvelet transforms are used for seismic dataset inpainting. Similar to (2.43), the BPDN method is used to find the sparse representation of the traces that are still available and the



(a) Original seismic dataset

(b) Noisy seismic dataset with 33% missing traces

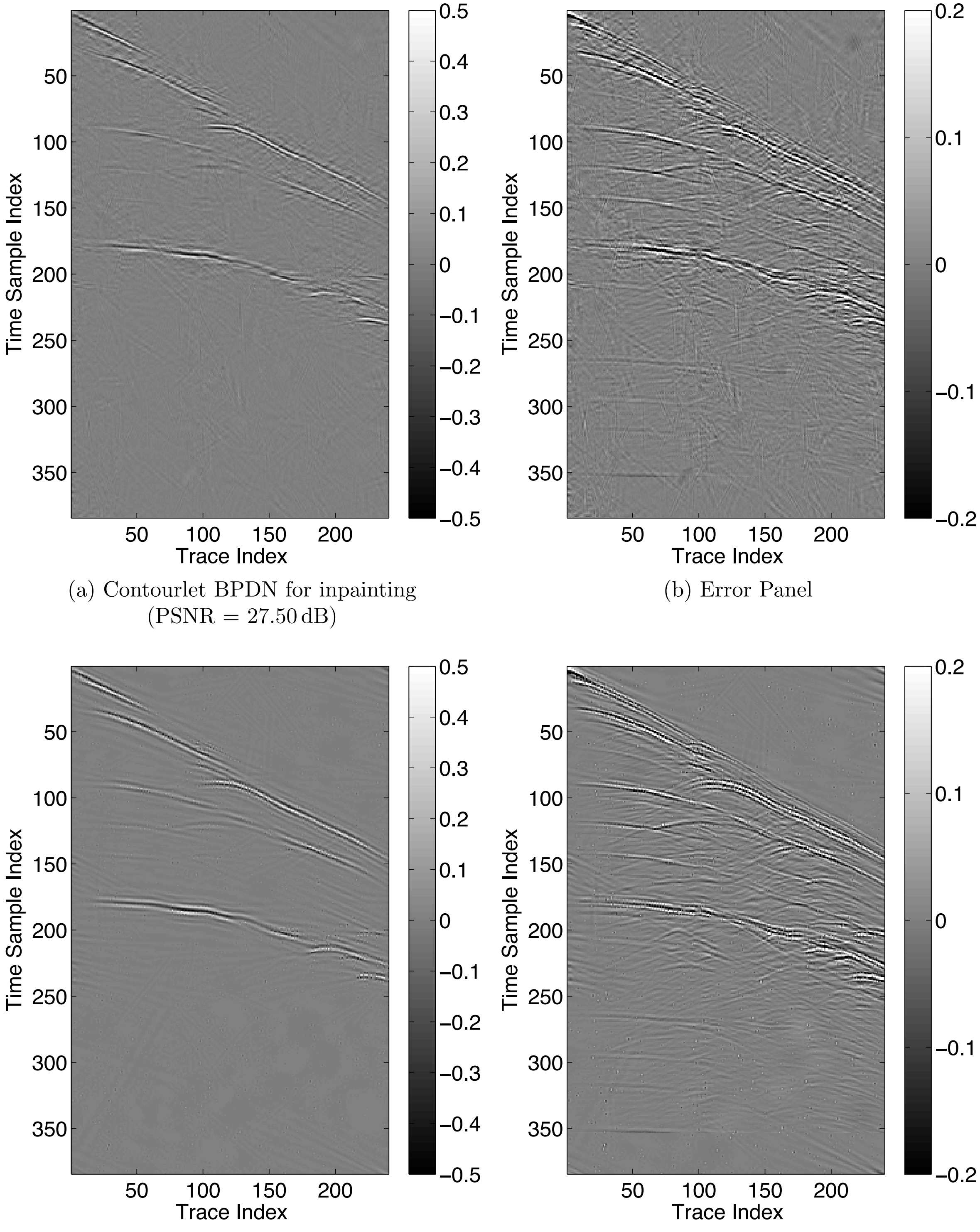
Figure 2.14: The original and noisy seismic dataset with 33% missing traces

missing traces can therefore be inferred via inverse transform operations as follows

$$\begin{cases} \hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\boldsymbol{\beta} \odot (\mathbf{w} - \boldsymbol{\Phi}\mathbf{x})\|_2^2 \leq \|\boldsymbol{\beta}\|_0 \cdot \sigma^2 \\ \hat{\mathbf{s}} = \boldsymbol{\Phi}\hat{\mathbf{x}} \end{cases} \tag{2.44}$$

where $\boldsymbol{\Phi}$ refers to the dictionary of the contourlet/curvelet synthesis operator. Figure 2.15 presents the inpainting results based on the BPDN method using the contourlet and curvelet transforms. The inpainting performance using the contourlets can achieve PSNR = 27.50 dB while using the curvelets can achieve PSNR = 28.12 dB. Still, just like in the pure denoising scenario, pseudo-Gibbs artifacts are quite obvious in the inpainting results.

Next, inpainting experiments are carried out, following the procedure in Algorithm 2.7. In the patch-based inpainting framework, one can fill the "holes" whose sizes are smaller than that of the atoms [82]. Therefore, in this case, the patch size is set to a slightly larger size $n_z \times n_x = 24 \times 24$, and a non-redundant DCT dictionary $\boldsymbol{\Phi}$ of size



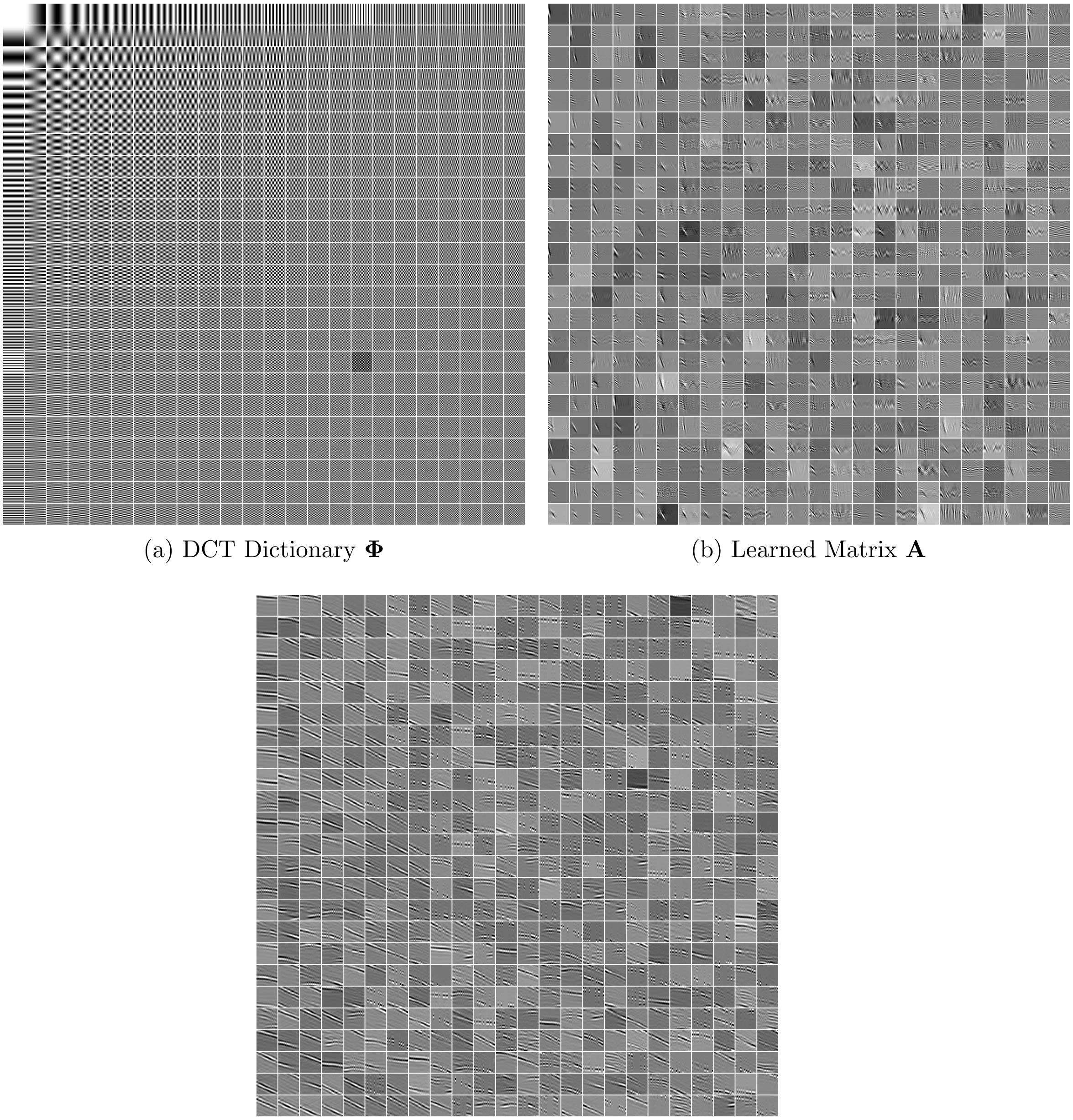
(a) Contourlet BPDN for inpainting (PSNR = 27.50 dB)

(b) Error Panel

(c) Curvelet BPDN for inpainting (PSNR = 28.12 dB)

(d) Error Panel

Figure 2.15: Inpainting results based on BPDN using the fixed multi-scale transforms without dictionary learning: (a) inpainted result by contourlet-based BPDN method (PSNR = 27.50 dB), (b) is the difference of (a) to the original data, (c) inpainted result by curvelet-based BPDN method (PSNR = 28.12 dB), and (d) is the difference between (c) and the original data.

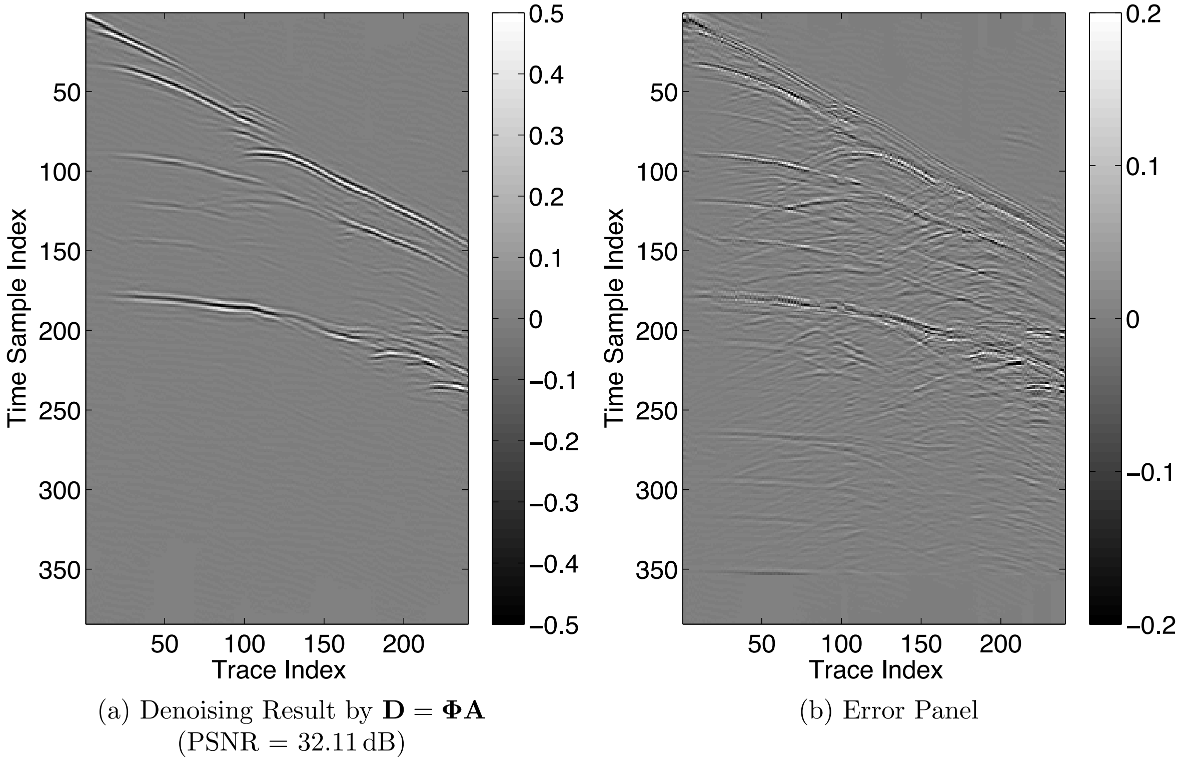


(a) DCT Dictionary $\mathbf{\Phi}$

(b) Learned Matrix $\mathbf{A}$

(c) Overall Learned Dictionary $\mathbf{D} = \mathbf{\Phi A}$

Figure 2.16: Base dictionary (DCT) and learned dictionaries



(a) Denoising Result by $\mathbf{D} = \mathbf{\Phi A}$ (PSNR = 32.11 dB)

(b) Error Panel

Figure 2.17: Inpainted result of dictionary learning method using DCT matrix as the base dictionary is shown in (a) and (b) is the difference of (a) to the original data.

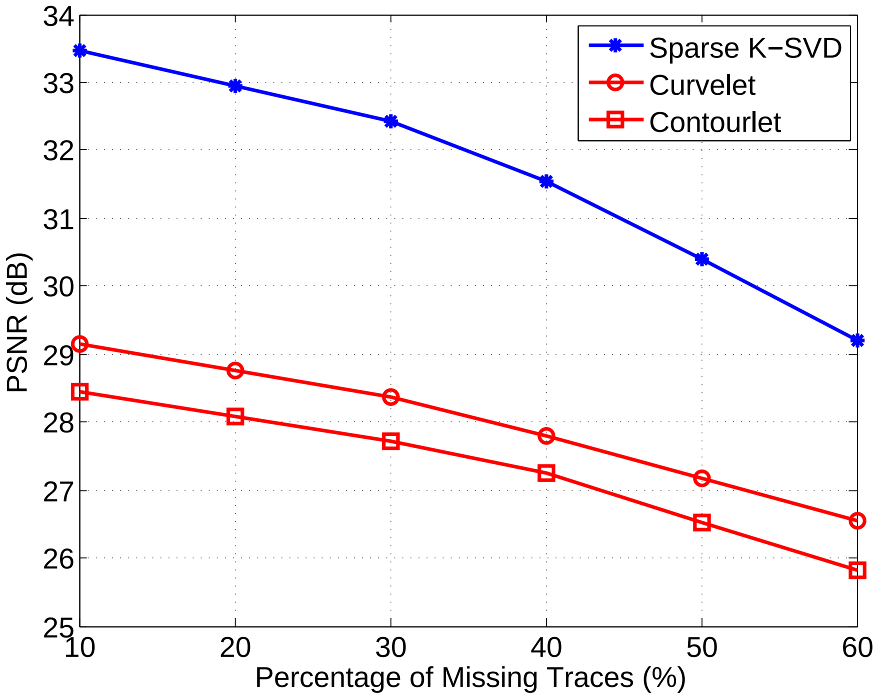


Figure 2.18: PSNR versus percentage of missing traces

$N \times N = 576 \times 576$ is selected as the base dictionary. Similarly, a total number of 10000 overlapping patches are randomly selected from the corrupted seismic dataset for dictionary learning and the sparse matrix is initialized to identity. The atom sparsity level p is set to 50. Figure 2.16(a) shows the non-redundant DCT base dictionary, while the learned sparse matrix $\mathbf{A}$ after $K = 20$ training iterations is visualized in Figure 2.16(b). The overall dictionary, $\mathbf{D} = \mathbf{\Phi A}$ of size 576×576, is visualized in Figure 2.16(c).

Based on this double-sparsity learned dictionry, the inpainting result can be obtained by (2.41). Its performance has been improved to PSNR = 32.11 dB, as shown in Figure 2.17(a), and the corresponding error panel is shown in Figure 2.17(b). Comparing to the contourlet and curvelet transforms, this result exhibits no pseudo-Gibbs artifacts around wave fronts. More experiments were performed in which the percentage of missing traces ranges from 10% to 60% and the PSNR performance curves are provided in Figure 2.18. The reconstruction result with dictionary learning method based on Algorithm 2.7 based on the sparse K-SVD algorithm yields much better PSNR values than fixed transforms.

CHAPTER III

EFFICIENT FULL WAVEFORM INVERSION BASED ON ONLINE ORTHONORMAL DICTIONARY LEARNING

3.1 Introduction

Full waveform inversion (FWI) is a data-fitting procedure that minimizes the misfit between recorded and calculated seismic data to create high-resolution quantitative subsurface medium models. A conventional FWI method is carried out iteratively. Each iteration consists of solving wave equations with the current model parameters to generate seismic data, calculating the value as well as the gradient of the misfit function, and updating the model parameters with an optimization method [65, 127, 128, 135]. The efficiency of these three components determines the industrial applicability of FWI. By recording the response of sequential sources on the surface or in the water, a wide-aperture seismic survey typically covers a large area of interest. Because the dimensionality of seismic datasets and models after finite difference discretization could be huge, computation of forward modeling, misfit calculation and model updating in FWI could be very intensive.

Reducing the computational cost of FWI has been an active research area for many years. When a frequency-domain FWI is carried out, one can divide the frequency range of interest into several bands, and invert only a few frequencies per band, sequentially from the low to high frequency bands, to help reduce the cost [18, 121]. Another well-known method for cost reduction is to generate simultaneous shots by linearly combining many different sequential shots at different source positions with random weights [8, 26, 64, 87], or randomly choosing a few sequential shots at each FWI iteration [75, 137].

Random source encoding or random sequential shots selection results in crosstalk noise or subsampling aliasing and introduces harmful artifacts in the inversion result. In order to alleviate these noisy artifacts, ℓ_1-norm sparsity regularization methods for compressive sensing (CS) can be applied into FWI by assuming sparsity of the velocity models over another domain. In previous research, these sparsity constraints have been imposed in a variety of fixed transform domains, such as wavelet [78], seislet [140] and curvelet [57, 73, 74].

This chapter [144] proposes a CS-based Gauss-Newton FWI framework in which the sparsity of model perturbations is exploited by a novel adaptive transform called the Sparse Orthonormal Transform (SOT). Unlike the traditional multi-scale transforms whose dictionaries are fixed and predefined as analytical functions, SOT is an adaptive transform whose dictionary is dynamically learned from a set of small patches extracted from model perturbations and hence can achieve sparser representations for the same kind of signals. Such sparsity promotion enables a significant reduction on the amount of data used in FWI.

The rest of this chapter is organized as follows. Section 3.2 reviews the Gauss-Newton method of FWI and explains why its computational complexity is intensive. Section 3.3 introduces the design process of SOT, including the efficient orthonormal dictionary learning method as well as its online approach for practical implementation in FWI problems. Definitions and operations that wrap a dictionary into a global transform, and empirical methods of parameter selection are also introduced here. Section 3.4 describes the randomization technique based on the linear property of wave equations, introduces the practical optimization method for solving the least-squares optimization method with ℓ_1-norm sparsity constraints, and summarizes the overall compressive FWI framework based on the SOT. Section 3.5 provides the numerical experiments of the proposed method on actual velocity models and compares the performance with other FWI frameworks that use the full data set for inversion, as well as methods that use compressive data but employ the curvelet transform for sparsity promotion.

3.2 Gauss-Newton method

As we have briefly introduced in Chapter 1, FWI aims to recover the model $\mathbf{m}$ whose forward modeling data $\boldsymbol{\mathcal{F}}(\mathbf{m})$ fits the recorded seismic data $\mathbf{d}_{\text{obs}}$. This can be formulated by minimizing the following misfit function

$$E(\mathbf{m}) \triangleq \frac{1}{2}\|\mathbf{d}_{\text{obs}} - \boldsymbol{\mathcal{F}}(\mathbf{m})\|_2^2. \tag{1.8 revisited}$$

FWI searches for the minimum of $E(\mathbf{m})$ in an iterative manner $\mathbf{m}_{k+1} = \mathbf{m}_k + \delta\mathbf{m}_k$, $k = 0, 1, 2, \dots$ where $\delta\mathbf{m}_k$ is the optimal model perturbation for each iteration. Solving for $\delta\mathbf{m}_k$ can be done with the second-order Taylor expansion of (1.8) in a small vicinity $\delta\mathbf{m}$ of $\mathbf{m}_k$

$$E(\mathbf{m}) = E(\mathbf{m}_k) + \delta\mathbf{m}^T\mathbf{g}_k + \frac{1}{2}\delta\mathbf{m}^T\mathbf{H}_k\delta\mathbf{m} + o(\|\delta\mathbf{m}\|^3) \tag{1.9 revisited}$$

where $\mathbf{g}_k \triangleq \dfrac{\partial E(\mathbf{m}_k)}{\partial \mathbf{m}}$ denotes the gradient of the misfit function $E(\mathbf{m})$ evaluated at $\mathbf{m}_k$ and $\mathbf{H}_k \triangleq \dfrac{\partial^2 E(\mathbf{m}_k)}{\partial \mathbf{m}^2}$ denotes the full Hessian matrix whose elements are the second-order partial derivatives of $E(\mathbf{m})$ at $\mathbf{m}_k$. By setting the gradient of $E(\mathbf{m})$ expressed in (1.9) with respect to $\delta\mathbf{m}$ be zero and ignoring the $o(\|\delta\mathbf{m}\|^3)$ term, $\delta\mathbf{m}_k$ satisfies

$$\mathbf{H}_k\delta\mathbf{m}_k = -\mathbf{g}_k. \tag{1.10 revisited}$$

The explicit expression for $\mathbf{g}_k$ is

$$\mathbf{g}_k \triangleq \frac{\partial E(\mathbf{m}_k)}{\partial \mathbf{m}} = -\Re\left\{\left[\frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{m}_k)}{\partial \mathbf{m}}\right]^\dagger (\mathbf{d}_{\text{obs}} - \boldsymbol{\mathcal{F}}(\mathbf{m}_k))\right\} = -\Re\left\{\mathbf{J}_k^\dagger \delta\mathbf{d}_k\right\} \tag{3.1}$$

where $\delta\mathbf{d}_k \triangleq \mathbf{d}_{\text{obs}} - \boldsymbol{\mathcal{F}}(\mathbf{m}_k)$, and $\mathbf{J}_k \triangleq \dfrac{\partial \boldsymbol{\mathcal{F}}(\mathbf{m}_k)}{\partial \mathbf{m}}$ is the Jacobian matrix of $\boldsymbol{\mathcal{F}}(\cdot)$ which indicates the sensitivity of the forward modeling data with respect to the model perturbation. The Hessian matrix $\mathbf{H}_k$ can be written as

$$\mathbf{H}_k \triangleq \frac{\partial^2 E(\mathbf{m}_k)}{\partial \mathbf{m}^2} = \Re\left\{\mathbf{J}_k^\dagger\mathbf{J}_k\right\} - \Re\left\{\left[\left(\frac{\partial \mathbf{J}_k^\dagger}{\partial m_1}\right)\delta\mathbf{d}_k, \cdots, \left(\frac{\partial \mathbf{J}_k^\dagger}{\partial m_N}\right)\delta\mathbf{d}_k\right]\right\}. \tag{3.2}$$

The method that inserts (3.1) and (3.2) into (1.10) for solving $\delta\mathbf{m}_k$ is referred to as Newton's method. However, the second term of $\mathbf{H}_k$ is small and hard to obtain [105, 129] and, therefore, is dropped most of the time. When $\mathbf{H}_k$ in (3.2) is expressed with only its first term $\mathbf{H}_k = \Re\left\{\mathbf{J}_k^\dagger \mathbf{J}_k\right\}$, (1.10) becomes the following normal equation

$$\left[\Re\left\{\mathbf{J}_k^\dagger \mathbf{J}_k\right\}\right]\delta\mathbf{m}_k = \Re\left\{\mathbf{J}_k^\dagger \delta\mathbf{d}_k\right\}. \tag{3.3}$$

The method that solves for $\delta\mathbf{m}_k$ with (3.3) is referred to as the Gauss-Newton method.

When $\mathbf{J}_k$ is full-rank, (3.3) has a unique solution $\delta\mathbf{m}_k$ that actually minimizes the following linear least-squares objective function

$$J_k(\delta\mathbf{m}) \triangleq \frac{1}{2}\|\delta\mathbf{d}_k - \mathbf{J}_k\delta\mathbf{m}\|_2^2. \tag{3.4}$$

Therefore, solving FWI with the Gauss-Newton method is equivalent to minimizing (3.4) for every iteration. Furthermore, comparing (3.4) with the LSRTM misfit in (1.6), one can see that each FWI iteration based on the Gauss-Newton method is equivalent to an LSRTM problem in which the currently estimated $\mathbf{m}_k$ is deemed to be the background model.

3.2.1 Computation of the Gradient and Hessian Matrix

In order to compute the gradient $\mathbf{g}_k$ and the Hessian matrix $\mathbf{H}_k$ for solving FWI, the forward modeling operator $\boldsymbol{\mathcal{F}}(\mathbf{m})$ needs to be specified. In the following discussion, the seismic wave propagation is modeled by the frequency-domain constant-density wave equation

$$\left(-m(\mathbf{x})\omega^2 - \nabla^2\right)\hat{p}(\mathbf{x};\omega,\mathbf{x}_s) = \hat{f}(\omega)\delta(\mathbf{x}-\mathbf{x}_s). \tag{1.11 revisited}$$

The model $\mathbf{m} \triangleq [m(\mathbf{x}_1),\ldots,m(\mathbf{x}_{N_zN_x})]^T$ is a vector of length N_zN_x for the model parameters, where N_z and N_x are the number of grid points in the vertical and lateral directions, respectively, i.e., the size of the model can also be regarded as $N_z \times N_x$ after 2D reshaping. It is reasonable to assume the source shot signature

$\hat{f}(\omega)$ is known and fixed. Given the frequency-domain Green's function $\hat{G}(\mathbf{x};\omega,\mathbf{x}_s)$ for the model $\mathbf{m}$ which is the solution of

$$\left(-m(\mathbf{x})\omega^2 - \nabla^2\right)\hat{G}(\mathbf{x};\omega,\mathbf{x}_s) = \delta(\mathbf{x}-\mathbf{x}_s), \tag{3.5}$$

the general solution of the wave equation (1.11) can be expressed as

$$\hat{p}(\mathbf{x};\omega,\mathbf{x}_s) = \hat{f}(\omega)\hat{G}(\mathbf{x};\omega,\mathbf{x}_s), \tag{3.6}$$

and the receivers collect $\hat{p}(\mathbf{x}_r;\omega,\mathbf{x}_s)$ at their locations $\mathbf{x}=\mathbf{x}_r$. Therefore, the operator $\boldsymbol{\mathcal{F}}(\mathbf{m})$ is defined to map the model $\mathbf{m}$ to a set of wavefield samples $\mathbf{d} \triangleq \{\hat{p}(\mathbf{x}_r;\omega,\mathbf{x}_s)\}$ collected at all receiver locations $\mathbf{x}_r$ for all source positions $\mathbf{x}_s$ and all frequencies ω.

The Jacobian matrix $\mathbf{J}_k \triangleq \dfrac{\partial\boldsymbol{\mathcal{F}}(\mathbf{m}_k)}{\partial\mathbf{m}}$ can be computed based on the Born approximation theory [51, 139], which regulates the wavefield perturbation $\widehat{\delta p}(\mathbf{x};\omega,\mathbf{x}_s)$ on the background wavefield $\hat{p}_k(\mathbf{x};\omega,\mathbf{x}_s)$ resulting from a small model perturbation $\delta m(\mathbf{x})$ on the background model $m_k(\mathbf{x})$. By taking the temporal Fourier transform on both sides of (1.2), the Born approximation of the frequency-domain wave equation (1.11) has the form

$$\left(-m_k(\mathbf{x})\omega^2 - \nabla^2\right)\widehat{\delta p}(\mathbf{x};\omega,\mathbf{x}_s) = \omega^2\delta m(\mathbf{x})\hat{p}_k(\mathbf{x};\omega,\mathbf{x}_s), \tag{3.7}$$

so that the solution collected at $\mathbf{x}=\mathbf{x}_r$ is

$$\widehat{\delta p}(\mathbf{x}_r;\omega,\mathbf{x}_s) = \omega^2\hat{f}(\omega)\sum_{\mathbf{x}\in\mathcal{U}}\delta m(\mathbf{x})G_k(\mathbf{x}_r;\omega,\mathbf{x})G_k(\mathbf{x};\omega,\mathbf{x}_s). \tag{3.8}$$

where the sum is taken over N_zN_x grid points $\mathbf{x}$ in the 2D subsurface medium $\mathcal{U}$ to take all physically acceptable scattering scenarios into account. Thus, for one specified source $\mathbf{x}_s$ and a single frequency ω, the (i,j)-th element of the Jacobian sub-matrix $\mathbf{J}_k(\omega,\mathbf{x}_s)$ that reflects the small wavefield change $\widehat{\delta p}(\mathbf{x}_{r_i};\omega,\mathbf{x}_s)$ at receiver location $\mathbf{x}_{r_i}$ due to a small model change $\delta m(\mathbf{x}_j)$ at location $\mathbf{x}_j$, is given by

$$[\mathbf{J}_k(\omega,\mathbf{x}_s)]_{ij} \triangleq \lim_{\delta m(\mathbf{x}_j)\to 0}\frac{\widehat{\delta p}(\mathbf{x}_{r_i};\omega,\mathbf{x}_s)}{\delta m(\mathbf{x}_j)} = \omega^2\hat{f}(\omega)G_k(\mathbf{x}_{r_i};\omega,\mathbf{x}_j)G_k(\mathbf{x}_j;\omega,\mathbf{x}_s). \tag{3.9}$$

The size of $\mathbf{J}_k(\omega, \mathbf{x}_s)$ is $N_r \times N_z N_x$ where N_r is the number of receivers. To obtain the entire Jacobian matrix $\mathbf{J}_k$, (3.9) is used repeatedly to determine $\mathbf{J}_k(\omega, \mathbf{x}_s)$ for all sources and frequencies of interest, i.e., N_s sources $\mathbf{x}_s \in \mathcal{S}$ and N_ω frequencies $\omega \in \Omega$. Finally, all of these different sub-matrices $\mathbf{J}_k(\omega, \mathbf{x}_s)$ are vertically concatenated to form the huge matrix $\mathbf{J}_k$ of size $N_\omega N_s N_r \times N_z N_x$ that can be used in the objective function (3.4). Inserting $\mathbf{J}_k$ back into the matrix-based expressions of the gradient $\mathbf{g}_k$ and the (approximate) Hessian matrix $\mathbf{H}_k$ yields the element-wise formulation

$$\begin{aligned} \mathbf{g}_k(\mathbf{x}_i) &= \left[-\Re\left\{\mathbf{J}_k^\dagger \delta\mathbf{d}_k\right\}\right]_i \\ &= -\Re\left\{\sum_{\omega\in\Omega} \omega^2 \hat{f}(\omega) \sum_{\mathbf{x}_s\in\mathcal{S}} \sum_{\mathbf{x}_r\in\mathcal{S}} G_k(\mathbf{x}_r;\omega,\mathbf{x}_i) G_k(\mathbf{x}_i;\omega,\mathbf{x}_s) \left(\overline{\delta d_k(\mathbf{x}_r;\omega,\mathbf{x}_s)}\right)\right\} \end{aligned} \tag{3.10}$$

and

$$\begin{aligned} \mathbf{H}_k(\mathbf{x}_i, \mathbf{x}_j) &= \left[\Re\left\{\mathbf{J}_k^\dagger \mathbf{J}_k\right\}\right]_{ij} \\ &= \Re\left\{\sum_{\omega\in\Omega} \omega^4 |\hat{f}(\omega)|^2 \sum_{\mathbf{x}_s\in\mathcal{S}} G_k(\mathbf{x}_i;\omega,\mathbf{x}_s)\overline{G_k(\mathbf{x}_j;\omega,\mathbf{x}_s)} \sum_{\mathbf{x}_r\in\mathcal{S}} G_k(\mathbf{x}_r;\omega,\mathbf{x}_i)\overline{G_k(\mathbf{x}_r;\omega,\mathbf{x}_j)}\right\}. \end{aligned} \tag{3.11}$$

3.2.2 Dimensionality Reduction Methods

From Equations (3.10) and (3.11), it is obvious that the complexity of the Gauss-Newton method comes primarily from the computation and inversion of the Hessian matrix $\mathbf{H}_k$. Unfortunately, due to the fact that $N_\omega N_s N_r$ and $N_z N_x$ are very large, it is prohibitive to compute $\mathbf{H}_k^{-1}$ directly with the entire data set in industrial-scale FWI problems. In order to reduce the computational complexity of FWI, it has been widely reported that for cases with a large acquisition aperture and wide frequency bandwidth, $\mathbf{H}_k$ is almost diagonally dominant, so $\mathbf{H}_k^{-1}$ can be further approximated with a diagonal matrix [10, 60, 97, 102, 106, 120].

The development of CS theories provides another perspective to lower the complexity of Gauss-Newton FWI by reducing the problem dimensionality rather than simplifying the Hessian matrix, when sparsity of the model can be exploited. This approach suggests that minimizing the linear least-squares objective function in (3.4) for each Gauss-Newton iteration can be replaced by the following optimization problem

$$\min_{\boldsymbol{\alpha}} \left\{ J_k^{(\mathrm{W})}(\boldsymbol{\alpha}) \triangleq \frac{1}{2}\|\mathbf{W}_k\delta\mathbf{d}_k - \mathbf{W}_k\mathbf{J}_k\boldsymbol{\mathcal{D}}(\boldsymbol{\alpha})\|_2^2 \right\} \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_1 \leq \tau_k \tag{3.12}$$

where $\mathbf{W}_k$ is a subsampling matrix for dimensionality reduction which can be different for each iteration k for better performance [57, 64, 73, 74, 137]; and the operator $\boldsymbol{\mathcal{D}}$ is a transform such that the model perturbation can be represented as $\delta\mathbf{m} = \boldsymbol{\mathcal{D}}(\boldsymbol{\alpha})$ with the coefficient vector $\boldsymbol{\alpha}$ being sparse.

Leaving the design of $\mathbf{W}_k$ aside for a while, a fundamental consideration in employing this representation of the model perturbation is the choice of the transform $\boldsymbol{\mathcal{D}}$. It is usually appealing to choose multiscale transforms such as wavelets, curvelets, seislets, etc. These fixed transforms have proven their analytical optimality for sparse representation of multidimensional signals with assumed features such as smooth lines or curves, and hence their success in applications relies on how suitable the signals in question fit the assumptions. In most cases, these multiscale transforms have efficient algorithmic implementations in the spatial-frequency domain and, as a result, their representations as dictionaries $\mathbf{D}$ are implicit. In the last several years, many authors [57, 73, 74, 78, 140] have developed methods that exploit the sparsity of $\delta\mathbf{m}$ by using various multiscale transforms to solve FWI problems efficiently.

The remainder of this chapter investigates how to exploit the sparsity of $\delta\mathbf{m}$ with a novel transform based on explicit adaptive dictionaries rather than implicit fixed dictionaries that exploit some assumed feature characteristics of the model. In particular, in each FWI iteration solving (3.12), a place is left for an adaptive transform that changes at each FWI iteration. The key to this approach is to infer explicit dictionary matrices $\mathbf{D}_k$ from a set of training examples and construct a transform $\boldsymbol{\mathcal{D}}_k$ based on these dictionaries that synthesizes $\delta\mathbf{m}$ from α. The similarity among different model perturbations suggests that small patches of previously optimized model perturbations $\{\delta\mathbf{m}_i\}_{i=0}^{k-1}$ could be an appropriate choice for a training set. The next section will discuss efficient dictionary learning algorithms that derive an adaptive dictionary from a set of training examples for sparse representation as well as the way to construct a transform operator based on this dictionary.

3.3 Sparse Orthonormal Transform

The CS technique can help to reduce the problem dimensionality of each Gauss-Newton problem in FWI, as long as the model perturbation $\delta\mathbf{m}$ is sparse with respect to some transform. Rather than directly applying fixed transforms based on off-the-shelf dictionaries such as wavelets, curvelets, seislets, etc., a novel kind of transform called the sparse orthonormal transform (SOT) is designed for sparsity promotion. The SOT is based on adaptive dictionaries that are learned from model perturbations to discover the inherent sparsity of $\delta\mathbf{m}$ at each FWI iteration.

As the probabilistic framework of dictionary learning suggests, given a matrix of R training examples $\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_R] \in \mathbb{R}^{N\times R}$, the dictionary learning method seeks the dictionary matrix $\mathbf{D} \in \mathbb{R}^{N\times L}$, $N \leq L$, that can represent $\mathbf{Y}$ with a set of sparse coefficients $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_R] \in \mathbb{R}^{L\times R}$. This process can be done by minimizing the following empirical cost function

$$e_R(\mathbf{Y}, \mathbf{D}) = \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \tag{3.13}$$

where λ is a Lagrange multiplier. The minimum of $e_R(\mathbf{Y}, \mathbf{D})$ can be found by a two-step iterative method in which the first step finds the sparse coefficients $\mathbf{X}$ given the fixed dictionary $\mathbf{D}$ and the second step updates the dictionary $\mathbf{D}$ given the sparse coefficients $\mathbf{X}$.

However, unlike the dictionary learning methods introduced in Chapter 2, two more features are incorporated into this case. First, the learned dictionaries are made orthonormal, i.e., $\mathbf{D} \in \mathbb{R}^{N\times N}$ and $\mathbf{D}^T\mathbf{D} = \mathbf{I}$, yielding a fast and straightforward alternating optimization scheme. Since an orthonormal dictionary is a Parseval frame, finding sparse coefficient vectors only requires simple matrix multiplication. On the other hand, an overcomplete or nonorthogonal dictionary loses this simplicity and makes sparse representation a complex pursuit problem. Second, the dictionaries are learned from training patches extracted from previously obtained model perturbations in an online manner, so that the iterative property of the Gauss-Newton method can be taken into account.

In this work, the R training examples $\mathbf{y}_i$, $i = 1, \ldots, R$, that form the matrix $\mathbf{Y}$ are extracted from patches of the optimized model perturbation $\delta\mathbf{m}_{k-1}$ obtained from the previous $(k-1)$-th FWI iteration. These patches cover all of $\delta\mathbf{m}_{k-1}$ and can be overlapping so that the matrix $\mathbf{D}$ will be a generative dictionary that provides sparse representations for all patches of $\delta\mathbf{m}$ in the following k-th FWI iteration. This updating strategy, which is called online learning, plays a critical role for iterative problems such as FWI.

3.3.1 Orthonormal Dictionary Learning

Imposing orthonormality on $\mathbf{D}$ provides a key property to solve the sparsity-constrained minimization problem so that the computational complexity of dictionary learning is greatly reduced. An efficient implementation of orthonormal dictionary learning has been successfully applied in natural image compression [115, 116] and seismic data denoising [19, 142]. With orthonormal dictionary learning and patch-based processing, the SOT can be designed based upon the previous results of $\delta\mathbf{m}$ without introducing significant extra computational complexity to FWI.

Orthonormal dictionary learning seeks the square dictionary matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ that minimizes the empirical cost function $e_R(\mathbf{Y}, \mathbf{D})$ defined in (3.13) with the orthonormality constraint $\mathbf{D}^T\mathbf{D} = \mathbf{I}$

$$\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_0 \quad \text{subject to} \quad \mathbf{D}^T\mathbf{D} = \mathbf{I}. \tag{3.14}$$

Note that the ℓ_0-norm sparsity constraint is used here for the hard-thresholding method that will be discussed next. Like the K-SVD and sparse K-SVD algorithms in the previous chapter, this problem can also be solved by using an iterative alternating optimization approach, but much more efficiently.

In each iteration, the first step is to find the sparsest representations of all columns of $\mathbf{Y} \in \mathbb{R}^{N\times R}$ over a fixed orthonormal dictionary $\mathbf{D}$, which is

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left(\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_0 \right). \tag{3.15}$$

If the hard-thresholding operator $\boldsymbol{\mathcal{H}}_\lambda(\cdot)$ with threshold λ is defined by

$$\boldsymbol{\mathcal{H}}_\lambda(\mathbf{X}) = \begin{cases} x_{ij}, & |x_{ij}| \geq \lambda \\ 0, & |x_{ij}| < \lambda, \end{cases} \tag{3.16}$$

then the solution to (3.15) given by

$$\hat{\mathbf{X}} = \boldsymbol{\mathcal{H}}_{\sqrt{\lambda}} \left(\mathbf{D}^T \mathbf{Y} \right) \tag{3.17}$$

is based on the following theorem.

Theorem 3.1 *For any given* $\mathbf{D} \in \mathbb{R}^{N\times N}$ *such that* $\mathbf{D}^T\mathbf{D} = \mathbf{I}$ *and* $\mathbf{y} \in \mathbb{R}^N$, *the minimization problem given by*

$$\min_{\mathbf{x}} \left(\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0 \right)$$

has a unique solution $\hat{\mathbf{x}} = \boldsymbol{\mathcal{H}}_{\sqrt{\lambda}} \left(\mathbf{D}^T \mathbf{y} \right)$.

Proof Since $\mathbf{D}^T\mathbf{D} = \mathbf{I}$, the objective function is rewritten as the sum of all components

$$\begin{aligned} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0 &= \left\|\mathbf{x} - \mathbf{D}^T\mathbf{y}\right\|_2^2 + \lambda \|\mathbf{x}\|_0 \\ &= \sum_{i=1}^{N} \left(\left(x_i - \mathbf{d}_i^T\mathbf{y}\right)^2 + \lambda |x_i|_0 \right) \end{aligned}$$

where $|x|_0 = 1$ if $x \neq 0$, and $|x|_0 = 0$ otherwise. For any $\mathbf{d}_i^T\mathbf{y}$,

$$\left(x_i - \mathbf{d}_i^T\mathbf{y}\right)^2 + \lambda |x_i|_0 = \begin{cases} \left(x_i - \mathbf{d}_i^T\mathbf{y}\right)^2 + \lambda, & \text{if } x_i \neq 0 \\ \left(\mathbf{d}_i^T\mathbf{y}\right)^2, & \text{if } x_i = 0 \end{cases}$$

and thus its minimum is the smaller value between λ and $\left(\mathbf{d}_i^T\mathbf{y}\right)^2$

$$\min_{x_i}\left(\left(x_i - \mathbf{d}_i^T\mathbf{y}\right)^2 + \lambda|x_i|_0\right) = \begin{cases} \lambda, & \text{if } \left|\mathbf{d}_i^T\mathbf{y}\right| \geq \sqrt{\lambda} \\ \left(\mathbf{d}_i^T\mathbf{y}\right)^2, & \text{if } \left|\mathbf{d}_i^T\mathbf{y}\right| < \sqrt{\lambda}. \end{cases}$$

The corresponding argument of the minimum is $\hat{x}_i = \mathbf{d}_i^T\mathbf{y}$ if $\left|\mathbf{d}_i^T\mathbf{y}\right| \geq \sqrt{\lambda}$, and $\hat{x}_i = 0$ otherwise. Using the compact notation of (3.16), $\hat{x}_i = \boldsymbol{\mathcal{H}}_{\sqrt{\lambda}}\left(\mathbf{d}_i^T\mathbf{y}\right)$ and, thus, for all components, $\hat{\mathbf{x}} = \boldsymbol{\mathcal{H}}_{\sqrt{\lambda}}\left(\mathbf{D}^T\mathbf{y}\right)$. ∎

The second step in solving (3.14) is to optimize the orthonormal dictionary $\mathbf{D}$ that minimizes the reconstruction error for a fixed matrix of sparse coefficients $\mathbf{X}$, i.e.,

$$\hat{\mathbf{D}} = \underset{\mathbf{D}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{subject to} \quad \mathbf{D}^T\mathbf{D} = \mathbf{I}. \tag{3.18}$$

Such a problem is called the "orthogonal Procrustes problem" [113]. The following theorem gives the solution to this problem.

Theorem 3.2 *Consider the problem* (3.18) *in which two matrices* $\mathbf{X}$, $\mathbf{Y} \in \mathbb{R}^{N\times R}$ *are given, define* $\mathbf{P} \triangleq \mathbf{X}\mathbf{Y}^T \in \mathbb{R}^{N\times N}$ *and denote its SVD as* $\mathbf{P} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ *where* $\mathbf{U}$, $\mathbf{V} \in \mathbb{R}^{N\times N}$ *are orthonormal matrices of singular vectors and* $\boldsymbol{\Sigma} \in \mathbb{R}^{N\times N}$ *is the diagonal matrix of singular values, then the orthonormal matrix* $\hat{\mathbf{D}} = \mathbf{V}\mathbf{U}^T \in \mathbb{R}^{N\times N}$ *is the solution.*

Proof

$$\begin{aligned} \because \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 &= \operatorname{Tr}\left((\mathbf{Y} - \mathbf{D}\mathbf{X})^T(\mathbf{Y} - \mathbf{D}\mathbf{X})\right) \\ &= \operatorname{Tr}\left(\mathbf{Y}^T\mathbf{Y}\right) - 2\operatorname{Tr}\left(\mathbf{Y}^T\mathbf{D}\mathbf{X}\right) + \operatorname{Tr}\left(\mathbf{X}^T\mathbf{D}^T\mathbf{D}\mathbf{X}\right) \\ &= \operatorname{Tr}\left(\mathbf{Y}^T\mathbf{Y}\right) + \operatorname{Tr}\left(\mathbf{X}^T\mathbf{X}\right) - 2\operatorname{Tr}\left(\mathbf{X}\mathbf{Y}^T\mathbf{D}\right) \end{aligned}$$

$$\therefore \underset{\mathbf{D}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 = \underset{\mathbf{D}}{\operatorname{argmax}} \operatorname{Tr}\left(\mathbf{X}\mathbf{Y}^T\mathbf{D}\right) \quad \text{subject to} \quad \mathbf{D}^T\mathbf{D} = \mathbf{I}$$

$$\begin{aligned}
\because \operatorname{Tr}\left(\mathbf{X}\mathbf{Y}^T\mathbf{D}\right) &= \operatorname{Tr}\left(\mathbf{P}\mathbf{D}\right) = \operatorname{Tr}\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{D}\right) = \operatorname{Tr}\left(\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{D}\mathbf{U}\right) \\
&= \sum_{i=1}^{N} \sigma_{ii}\mathbf{v}_i^T\mathbf{D}\mathbf{u}_i \\
&\leq \sum_{i=1}^{N} \sigma_{ii}
\end{aligned}$$

since $\sigma_{ii} > 0$ and $\left|\mathbf{v}_i^T\mathbf{D}\mathbf{u}_i\right| \leq 1$,

$$\therefore \max_{\mathbf{D}} \operatorname{Tr}\left(\mathbf{X}\mathbf{Y}^T\mathbf{D}\right) = \sum_{i=1}^{N} \sigma_{ii}.$$

The corresponding argument of the maximum is $\hat{\mathbf{D}} = \mathbf{V}\mathbf{U}^T$. ■

The orthonormal dictionary $\mathbf{D}$ can thus be learned by alternating between the above two steps iteratively until the cost function $e_R(\mathbf{Y}, \mathbf{D})$ is reduced to a limiting value. For each learning iteration, orthonormal dictionary learning needs three matrix multiplications that cost $\mathcal{O}(2RN^2 + N^3)$ and one SVD operation that costs $\mathcal{O}(N^3)$ to obtain both the sparse coding and the updated dictionary. Specifically, assume that the model size is denoted by $N_z \times N_x$ and the training patch size is denoted by $n_z \times n_x$, where $n_z \ll N_z$, $n_x \ll N_x$, and $N = n_z n_x$. If all possible overlapping patches are used for training, then the number of training patches $R = (N_z - n_z + 1)(N_x - n_x + 1)$, and each training iteration costs $\mathcal{O}((n_z n_x)^2(N_z - n_z + 1)(N_x - n_x + 1) + (n_z n_x)^3)$. The number of training iterations does not depend on these sizes, hence the overall complexity does not change in the sense of the Big-$\mathcal{O}$ notation. The foregoing analysis motivates the fact that the patch size should be small for dictionary learning algorithms; otherwise, the complexity would grow dramatically if n_z or n_x were large.

Based on Theorems 3.1 and 3.2, the orthonormal dictionary learning method can find all sparse representations and update all dictionary atoms in one pass. On the contrary, the overcomplete or nonorthogonal dictionary learning methods introduced in Chapter 2 have to invoke computationally expensive processes such as MP, BPDN or LASSO to sequentially modify sparse representations and dictionary atoms one by one. Therefore, the computational complexity of orthonormal dictionary learning method is significantly less than the others.

3.3.2 Dictionary-based Block-wise Transform

Dictionary learning methods use patches to form dictionaries and, therefore, the learned dictionary can only be applied on the patches rather than on the whole image. Previously, dictionary learning was used in *nearly-local* problems such as signal denoising or inpainting discussed in Chapter 2 where patches can be independently processed one by one. In the FWI problem, it is necessary to recover the entire model perturbation $\delta\mathbf{m}$ from compressive measurements. This, however, is a *global* problem where the compressive measurements encode the whole $\delta\mathbf{m}$ and thus all patches of $\delta\mathbf{m}$ need to be recovered at once. As a result, an invertible transform that can be applied to the whole $\delta\mathbf{m}$ is required. This subsection shows how to convert the local dictionaries $\mathbf{D}$ into a global transform $\boldsymbol{\mathcal{D}}$ that can be applied on the whole domain of $\delta\mathbf{m}$, and such a transform is named the sparse orthonormal transform (SOT).

The whole model perturbation $\delta\mathbf{m}$ can be exactly represented as

$$\delta\mathbf{m} = \boldsymbol{\mathcal{T}}^{-1} \sum_{(i,j)\in\mathcal{P}} \boldsymbol{\mathcal{R}}_{ij}^{\dagger} \left(\boldsymbol{\mathcal{R}}_{ij}\left(\delta\mathbf{m}\right)\right) \tag{3.19}$$

where the operator $\boldsymbol{\mathcal{R}}_{ij}$ extracts the (i, j)-th patch of size $N = n_z n_x$ from $\delta\mathbf{m}$, its adjoint $\boldsymbol{\mathcal{R}}_{ij}^{\dagger}$ tiles the (i, j)-th patch of size $N = n_z n_x$ back to $\delta\mathbf{m}$, and $\mathcal{P}$ refers to an index set of the selected patches that fully cover $\delta\mathbf{m}$. The averaging operator $\boldsymbol{\mathcal{T}} \triangleq \sum_{(i,j)\in\mathcal{P}} \boldsymbol{\mathcal{R}}_{ij}^{\dagger}\boldsymbol{\mathcal{R}}_{ij}$ is an invertible diagonal matrix so that $\boldsymbol{\mathcal{T}}^{-1}$ is a grid-by-grid (pixel-by-pixel) operation. Every block $\boldsymbol{\mathcal{R}}_{ij}\left(\delta\mathbf{m}\right) \in \mathbb{R}^N$ has a sparse representation $\boldsymbol{\alpha}_{ij} \in \mathbb{R}^N$ over a learned orthonormal dictionary $\mathbf{D} \in \mathbb{R}^{N\times N}$, i.e., $\boldsymbol{\mathcal{R}}_{ij}\left(\delta\mathbf{m}\right) = \mathbf{D}\boldsymbol{\alpha}_{ij}$, so the above representation of $\delta\mathbf{m}$ can be written as

$$\delta\mathbf{m} = \boldsymbol{\mathcal{T}}^{-1} \sum_{(i,j)\in\mathcal{P}} \boldsymbol{\mathcal{R}}_{ij}^{\dagger} \left(\mathbf{D}\boldsymbol{\alpha}_{ij}\right). \tag{3.20}$$

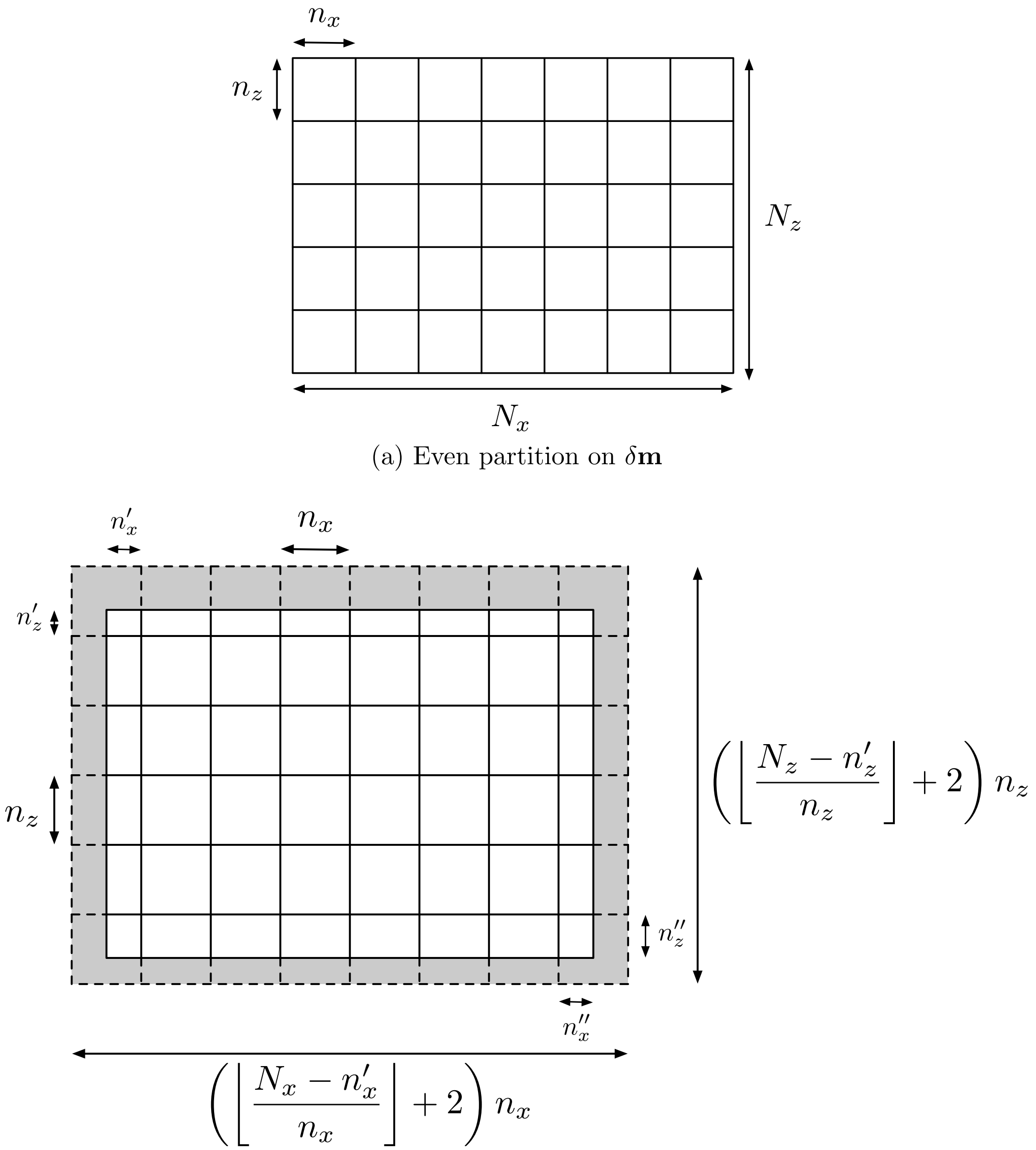
Since $\boldsymbol{\alpha}_{ij}$ has the same length as $\boldsymbol{\mathcal{R}}_{ij}\left(\delta\mathbf{m}\right)$, $\boldsymbol{\alpha}_{ij}$ fits into the same (i,j)-th patch of a global SOT coefficient $\boldsymbol{\alpha}$ because $\boldsymbol{\alpha}_{ij} = \boldsymbol{\mathcal{R}}_{ij}(\boldsymbol{\alpha})$. Therefore, the invertible SOT can be expressed as

$$
\begin{aligned}
\delta\mathbf{m} &= \boldsymbol{\mathcal{T}}^{-1} \sum_{(i,j)\in\mathcal{P}} \boldsymbol{\mathcal{R}}_{ij}^{\dagger}\left(\mathbf{D}\boldsymbol{\mathcal{R}}_{ij}(\boldsymbol{\alpha})\right) = \left[\boldsymbol{\mathcal{T}}^{-1} \sum_{(i,j)\in\mathcal{P}} \boldsymbol{\mathcal{R}}_{ij}^{\dagger}\mathbf{D}\boldsymbol{\mathcal{R}}_{ij}\right](\boldsymbol{\alpha}) = \boldsymbol{\mathcal{D}}(\boldsymbol{\alpha}) \\
\boldsymbol{\alpha} &= \boldsymbol{\mathcal{T}}^{-1} \sum_{(i,j)\in\mathcal{P}} \boldsymbol{\mathcal{R}}_{ij}^{\dagger}\left(\mathbf{D}^{T}\boldsymbol{\mathcal{R}}_{ij}(\delta\mathbf{m})\right) = \left[\boldsymbol{\mathcal{T}}^{-1} \sum_{(i,j)\in\mathcal{P}} \boldsymbol{\mathcal{R}}_{ij}^{\dagger}\mathbf{D}^{T}\boldsymbol{\mathcal{R}}_{ij}\right](\delta\mathbf{m}) = \boldsymbol{\mathcal{D}}^{\dagger}(\delta\mathbf{m})
\end{aligned}
\tag{3.21}
$$

where $\boldsymbol{\mathcal{D}} \triangleq \boldsymbol{\mathcal{T}}^{-1} \sum_{(i,j)\in\mathcal{P}} \boldsymbol{\mathcal{R}}_{ij}^{\dagger}\mathbf{D}\boldsymbol{\mathcal{R}}_{ij}$ is the global SOT synthesis (i.e., inverse transform) operator. The operator $\boldsymbol{\mathcal{D}}$ decomposes the global coefficients $\boldsymbol{\alpha}$ into blocks, reconstructs all the blocks into model patches with $\mathbf{D}$, and tiles the patches back to $\delta\mathbf{m}$ at the correct positions. Its adjoint operator $\boldsymbol{\mathcal{D}}^{\dagger} \triangleq \boldsymbol{\mathcal{T}}^{-1} \sum_{(i,j)\in\mathcal{P}} \boldsymbol{\mathcal{R}}_{ij}^{\dagger}\mathbf{D}^{T}\boldsymbol{\mathcal{R}}_{ij}$ is the global SOT analysis operator (i.e., transform) that decomposes the whole model perturbation $\delta\mathbf{m}$ into patches, converts all the patches into coefficient blocks with $\mathbf{D}^{T}$, and concatenates them into the global coefficient vector $\boldsymbol{\alpha}$.

In the design process of the SOT operators $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{D}}^{\dagger}$, it is usually preferred to have an index set $\mathcal{P}$ such that all selected patches are non-overlapping. As each patch $\boldsymbol{\mathcal{R}}_{ij}\left(\delta\mathbf{m}\right)$ has its own independent sparse representation coefficient $\boldsymbol{\alpha}_{ij}$ over the dictionary $\mathbf{D}$, a large number of overlapping patches would introduce too many degrees of freedom in the global SOT coefficient vector $\boldsymbol{\alpha}$, compromise its sparsity level, and hence impair the reconstruction of $\delta\mathbf{m}$. Similar to the computational complexity analysis for orthonormal dictionary learning, with the model size being $N_z \times N_x$ and the patch size being $n_z \times n_x$, the computational complexity of applying the SOT, or the inverse SOT, is $\mathcal{O}(n_z n_x N_z N_x)$ since each patch transform costs $\mathcal{O}((n_z n_x)^2)$ and there are about $(N_z N_x)/(n_z n_x)$ non-overlapping model patches.

There might be an issue when $\delta\mathbf{m}$ of size $N_z \times N_x$ cannot be evenly decomposed into small patches of size $n_z \times n_x$. In order to comply with the prerequisite of $\mathcal{P}$ that all selected patches should fully cover $\delta\mathbf{m}$, some patches along the boundary need to be treated with further caution, which is discussed as follows.


n_x
n_z
N_z
N_x
n'_x
n_x
n'_z
n_z
$\left(\left\lfloor \frac{N_z - n'_z}{n_z} \right\rfloor + 2 \right) n_z$
n''_z
n''_x
$\left(\left\lfloor \frac{N_x - n'_x}{n_x} \right\rfloor + 2 \right) n_x$

(a) Even partition on $\delta\mathbf{m}$

(b) Uneven partition on $\delta\mathbf{m}$

Figure 3.1: Different non-overlapping and covering partition schemes on $\delta\mathbf{m}$

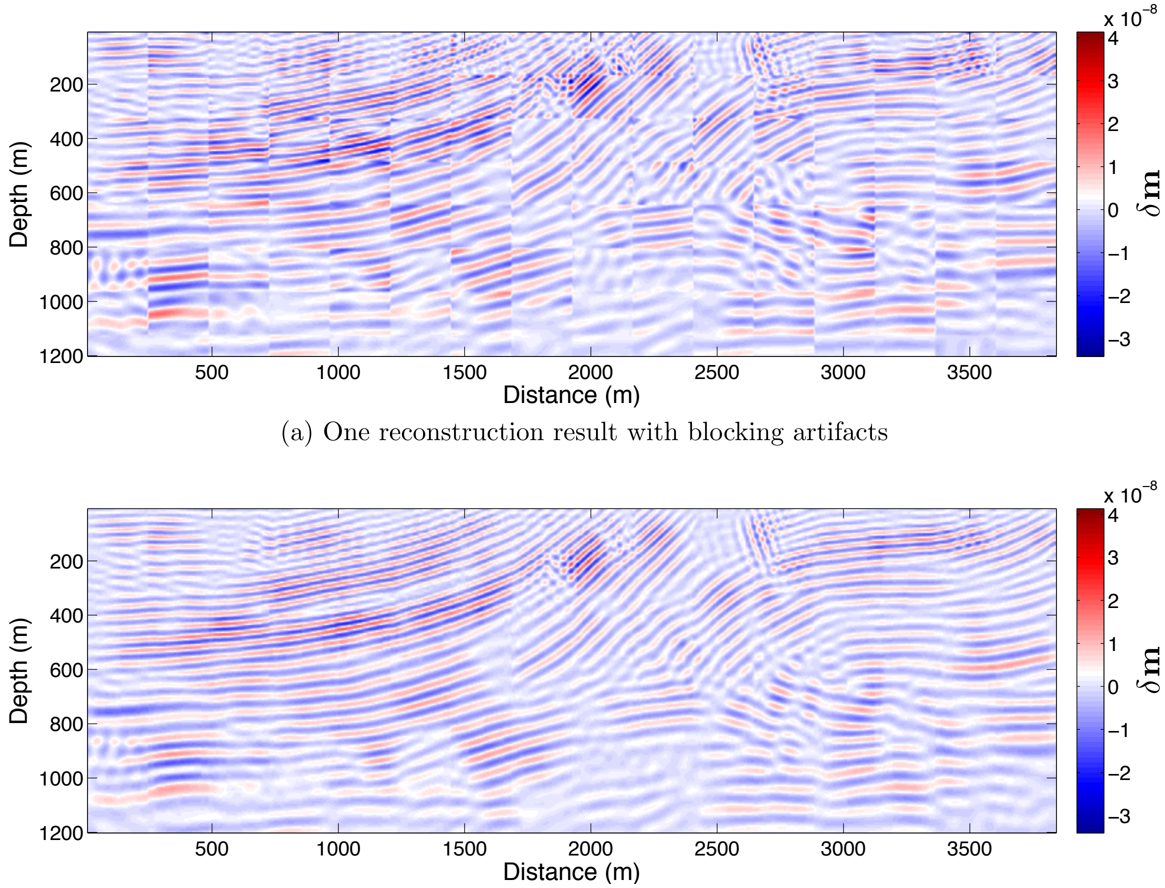
Figure 3.1 depicts two different partition schemes where non-overlapping patches fully cover $\delta\mathbf{m}$. In Figure 3.1(a), an even partition scheme is used when N_z, N_x are divisible by n_z, n_x, respectively. Figure 3.1(b) illustrates an uneven partition scheme such that the top, bottom, left and right boundary patches, and four corner patches, have different sizes smaller than $n_z \times n_x$, which is the size of the interior blocks. Note that the uneven partition scheme is more general than its even counterpart as it can be used in all cases, no matter whether $n_z \mid N_z$ and $n_x \mid N_x$, or not.

Since all patches are aligned in both horizontal and vertical directions, the size of the top-left corner patch determines the overall partition scheme as well as the index set $\mathcal{P}$, and hence can be denoted as $n'_z \times n'_x$ such that $n'_z \leq n_z$ and $n'_x \leq n_x$. Based on this definition, the top boundary patches are of size $n'_z \times n_x$ and the left boundary patches are of size $n_z \times n'_x$. If the size of the bottom-right corner patch is defined as $n''_z \times n''_x$, it is easy to find that $n''_z = \left((N_z - n'_z) - \left\lfloor \frac{N_z - n'_z}{n_z} \right\rfloor n_z \right)$ and $n''_x = \left((N_x - n'_x) - \left\lfloor \frac{N_x - n'_x}{n_x} \right\rfloor n_x \right)$ where $\lfloor \cdot \rfloor$ rounds a real number to its largest previous integer, and thus the bottom boundary patches have size $n''_z \times n_x$, the right boundary patches have size $n_z \times n''_x$, the top-right corner patch has size $n'_z \times n''_x$ and the bottom-left corner patch has size $n''_z \times n'_x$. Therefore, at most 9 kinds of patches with different sizes could be extracted in the uneven partition scheme. One would possibly seek a solution that could train at most 9 kinds of dictionaries for patches of different sizes. However, this is actually unnecessary since training so many different kinds of dictionaries would be quite expensive.

A much simpler solution, without the need of training multiple dictionaries with different sizes, would be padding zeros on the boundary of $\delta\mathbf{m}$ until it can be evenly decomposed. Such a zero-padding operation can be implicitly incorporated into the global SOT analysis operator $\boldsymbol{\mathcal{D}}^\dagger$. Taking Figure 3.1(b) as an example, $\boldsymbol{\mathcal{D}}^\dagger$ first pads zeros on the boundary of $\delta\mathbf{m}$ (the gray area) and expands its size to

$$\left(\left(\left\lfloor \frac{N_z - n'_z}{n_z} \right\rfloor + 2 \right) n_z \right) \times \left(\left(\left\lfloor \frac{N_x - n'_x}{n_x} \right\rfloor + 2 \right) n_x \right).$$

Then it is possible to evenly decompose the zero-padded $\delta\mathbf{m}$ into patches, all of size $n_z \times n_x$, convert all patches into coefficients with $\mathbf{D}^T$ and concatenate all coefficients into the global coefficient vector $\boldsymbol{\alpha}$. For the adjoint, the global SOT synthesis operator $\boldsymbol{\mathcal{D}}$ decomposes the global coefficient $\boldsymbol{\alpha}$ into blocks, reconstructs all blocks into model patches with $\mathbf{D}$, tiles all reconstructed patches back to the zero-padded $\delta\mathbf{m}$, and finally, removes its zero-padding area to recover the original size of $N_z \times N_x$.



(a) One reconstruction result with blocking artifacts

(b) Averaged reconstruction results with 6 different $\mathcal{P}$ settings for blocking artifact alleviation

Figure 3.2: A reconstruction of $\delta\mathbf{m}$ with only 1% of the coefficients in $\boldsymbol{\alpha}$

A global reconstruction of $\delta\mathbf{m}$ by tiling all its non-overlapping patches recovered from compressive measurements would lead to visible blocking artifact. An uneven partition scheme provides a solution that can mitigate this issue. Since $\mathcal{P}$ can be chosen freely now with flexible top-left corner patch sizes $n'_z \times n'_x$, one can reconstruct multiple $\delta\mathbf{m}$ with different $\mathcal{P}$ settings and average these results into one for blocking artifact alleviation.

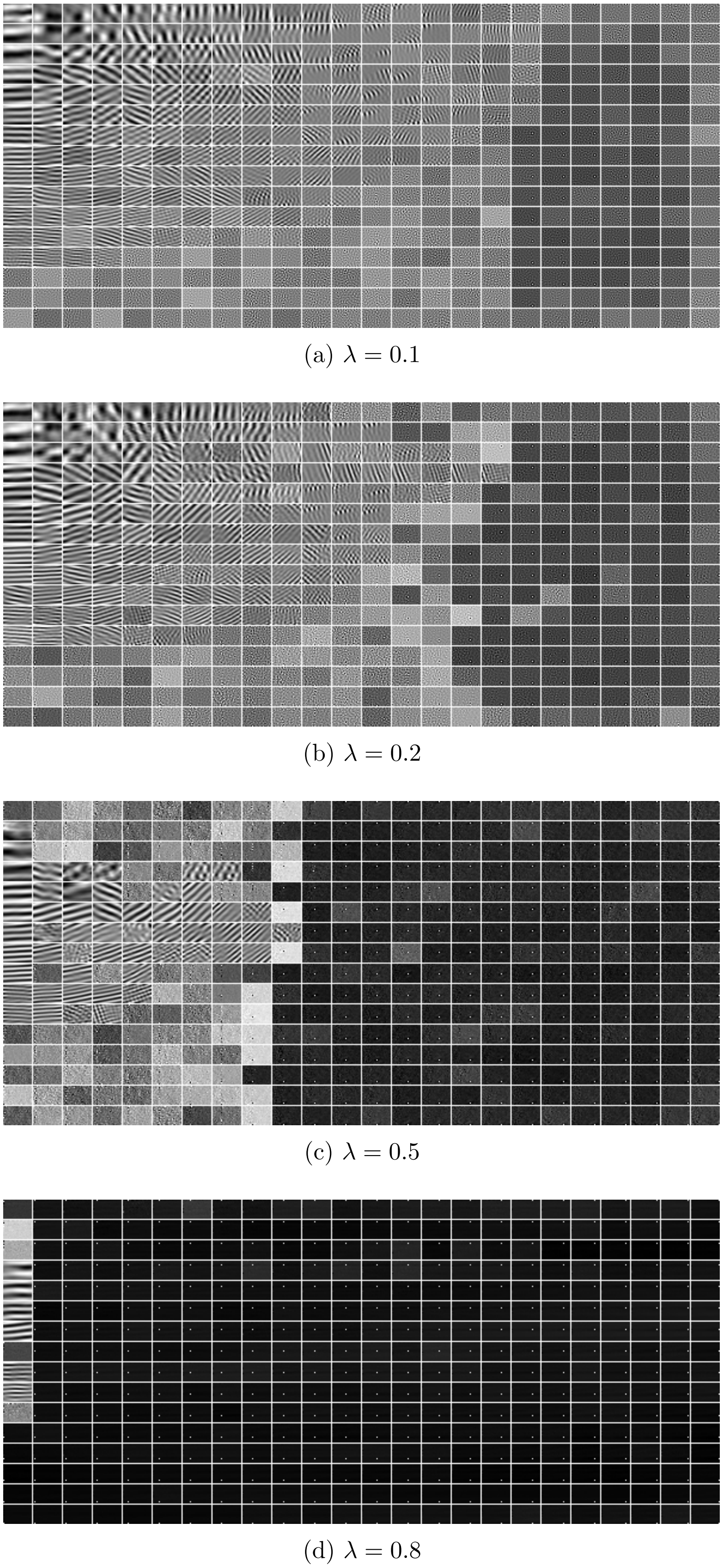
Figure 3.2 compares two reconstructed $\delta\mathbf{m}$ with only 1% of the coefficients in $\boldsymbol{\alpha}$, in which one exhibits clearly visible blocking artifacts while another one does not. The result shown in Figure 3.2(a) is affected by blocking artifacts because only one partition scheme has been used. These blocking artifacts are alleviated after six $\delta\mathbf{m}$ are reconstructed with different $\mathcal{P}$ settings and then averaged together, as shown in Figure 3.2(b).

3.3.3 Choice of Lagrange Parameter λ

The value of the Lagrange parameter λ in orthonormal dictionary learning (3.14) controls sparsity because it determines the design of dictionaries for a particular sparsity level and shapes the atoms of $\mathbf{D}$, as transform coefficients with absolute values smaller than $\sqrt{\lambda}$ are hard-thresholded to zero. A small λ would yield marginal change of $\mathbf{D}$ after each iteration since most elements in $\mathbf{C} = \mathbf{D}^T\mathbf{Y}$ would remain unchanged for $\hat{\mathbf{X}}$. The extreme case is when $\lambda = 0$, then $\hat{\mathbf{X}} = \mathbf{C} = \mathbf{D}^T\mathbf{Y}$, and it is trivial to solve (3.18) to obtain $\hat{\mathbf{D}} = \mathbf{D}$ which does not change at all. On the contrary, if λ were large, then most elements in $\mathbf{C} = \mathbf{D}^T\mathbf{Y}$ would be hard-thresholded to zeros for $\hat{\mathbf{X}}$, and $\mathbf{P} = \hat{\mathbf{X}}\mathbf{Y}^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ would be a low-rank matrix, resulting in many atoms in $\hat{\mathbf{D}} = \mathbf{V}\mathbf{U}^T$ resembling the trivial standard basis. The extreme case is when $\lambda = 1$, giving rank$(\mathbf{P}) = 0$, and $\hat{\mathbf{D}}$ degrades to $\mathbf{I}$. Some examples of $\mathbf{D} \in \mathbb{R}^{384\times384}$ learned with different values of λ are shown in Figure 3.3, where each atom of $\mathbf{D}$ is reshaped into a 2D block of size 16×24 for visualization. For $\lambda = 0.1$ and $\lambda = 0.2$ in Figures 3.3(a) and 3.3(b), many of the dictionary atoms exhibit directional characteristics. On the other hand, when $\lambda = 0.8$ in Figure 3.3(d) almost all of the dictionary atoms have a single nonzero value, i.e., the trivial basis.

Nonlinear approximation (NLA) can be used to verify the sparse representation capability of the learned orthonormal dictionary $\mathbf{D}$ (and the global SOT synthesis operator $\boldsymbol{\mathcal{D}}$) for a $\delta\mathbf{m}$. The NLA test keeps the l largest-magnitude coefficients from $\boldsymbol{\alpha}$ as $\widetilde{\boldsymbol{\alpha}}$, and then evaluates the normalized mean square error (NMSE) of the reconstruction

$$\mathrm{NMSE}(\delta\mathbf{m}; \boldsymbol{\mathcal{D}}, l) = 1 - \left\| \frac{\delta\mathbf{m} - \boldsymbol{\mathcal{D}}(\widetilde{\boldsymbol{\alpha}})}{\delta\mathbf{m} - \mathrm{mean}(\delta\mathbf{m})} \right\|_2^2, \tag{3.22}$$



(a) $\lambda = 0.1$

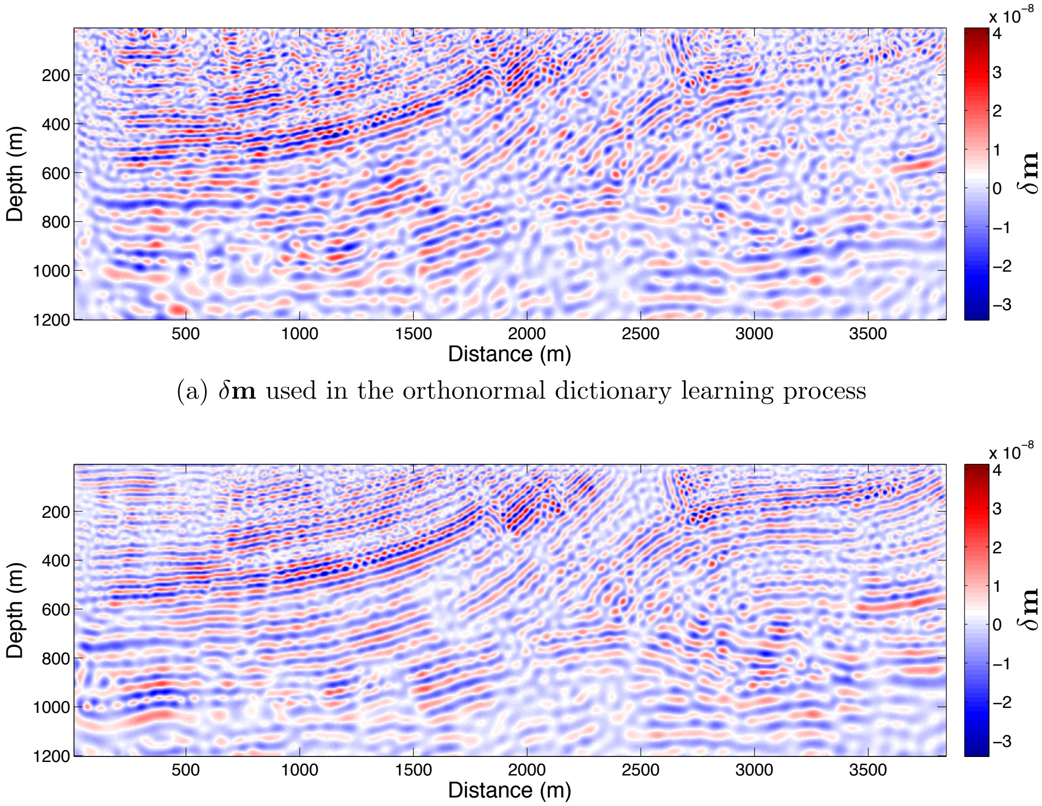
(b) $\lambda = 0.2$

(c) $\lambda = 0.5$

(d) $\lambda = 0.8$

Figure 3.3: Dictionaries $\mathbf{D} \in \mathbb{R}^{384\times384}$ trained with different values of λ

which varies from $-\infty$ (bad fit) to 1 (perfect fit).

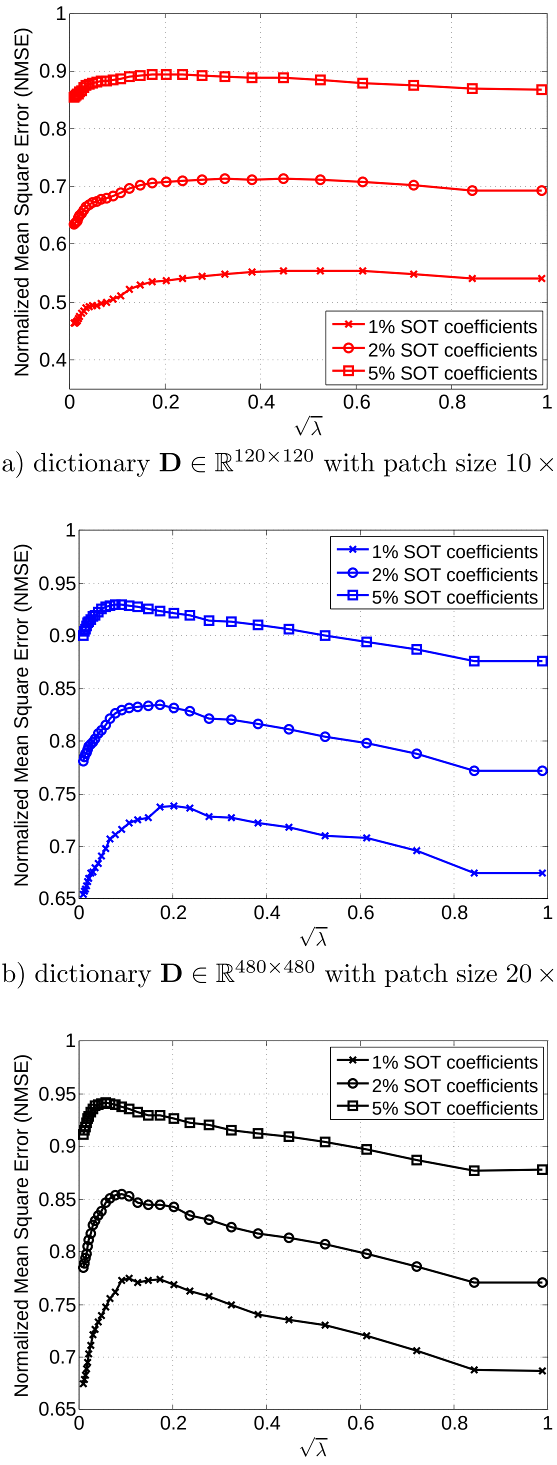


(a) $\delta\mathbf{m}$ used in the orthonormal dictionary learning process

(b) $\delta\mathbf{m}$ used for NLA test

Figure 3.4: Two model perturbations $\delta\mathbf{m}$ extracted from consecutive FWI iterations and used for orthonormal dictionary learning and the NLA test

The Lagrange parameter λ is usually related to an approximate noise level if dictionary learning is applied in a denoising problem [44]. However, its selection in a CS-based sparsity recovery problem still remains an open problem [122] so that λ is chosen empirically. A simplified experiment can be conducted to compare the NLA performance of learned orthonormal dictionaries $\mathbf{D}$ trained with different values of λ. Different dictionaries $\mathbf{D}$ are learned from training patches of different sizes extracted from a training model perturbation shown in Figure 3.4(a) and tested on a testing model perturbation shown in Figure 3.4(b). The NLA performance curves that indicate the relationship between NMSE and λ for different sparsity levels l (1%, 2% and 5% of largest-magnitude coefficients) are shown in Figure 3.5. In Figures



(a) dictionary $\mathbf{D} \in \mathbb{R}^{120\times120}$ with patch size 10×12

(b) dictionary $\mathbf{D} \in \mathbb{R}^{480\times480}$ with patch size 20×24

(c) dictionary $\mathbf{D} \in \mathbb{R}^{960\times960}$ with patch size 30×32

Figure 3.5: NLA performance curves of keeping 1%, 2% and 5% largest-magnitude coefficients for different patch sizes

3.5(a) to 3.5(c), the optimal λ that yields the highest NMSE depends on both the sparsity level l and the patch size $n_z \times n_x$, and the optimal λ tends to decrease if either the sparsity level l or the patch size $n_z \times n_x$ increases. These results indicate that $\lambda \in (0.15^2, 0.25^2)$ is expected to deliver good reconstructions for reasonable sparsity levels and patch sizes.

3.3.4 Online Orthonormal Dictionary Learning

The above orthonormal dictionary learning algorithm takes the training patch set as a whole so that a dictionary $\mathbf{D}$ could be learned offline and would remain static as a sparse representation. Generally speaking, such an offline approach cannot effectively handle very large training sets, or dynamic training sets that vary over time. In practice, FWI is an iterative problem where the optimized $\delta\mathbf{m}_k$ that offers training patches is changing over iterations. Therefore, to exploit the availability of new training patches from $\delta\mathbf{m}_k$, an online approach is proposed for orthonormal dictionary learning by minimizing the following expected cost function

$$e(\mathbf{D}) \triangleq \mathbb{E}_{\mathbf{y}} \left[\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0 \right] = \lim_{R\to\infty} e_R(\mathbf{Y}, \mathbf{D}) \quad \text{almost surely.} \tag{3.23}$$

Rather than spending too much effort on accurately minimizing the empirical cost function $e_R(\mathbf{Y}, \mathbf{D})$ in (3.13), [16] suggest minimizing $e(\mathbf{D})$ since $e_R(\mathbf{Y}, \mathbf{D})$ is merely an approximation of $e(\mathbf{D})$. Minimizing $e(\mathbf{D})$ does not rely on the number of patches R, but instead on the (unknown) stochastic characteristics of the training patches. The online approach learns a new dictionary $\mathbf{D}_k$ every time a new $\delta\mathbf{m}_{k-1}$ is ready, and the sequence of learned dictionaries can adapt to the variations of patches in later iterations.

Algorithm 3.1 summarizes a general version of the online orthonormal dictionary learning method in which the training examples $\mathbf{y}$ are drawn from a data stream source. In particular, at the end of the $(k-1)$-th FWI iteration, a batch of R training patches, each of size $n_z \times n_x$, are extracted from $\delta\mathbf{m}_{k-1}$ and normalized

```
Input: a data source from which input data $\mathbf{y} \in \mathbb{R}^N$ are drawn, initial orthonormal
       dictionary $\mathbf{D}_0 \in \mathbb{R}^{N\times N}$, Lagrange multiplier $\lambda$, number of update iterations
       $T$, mini-batch size $R$, Cauchy's convergence error bound $\epsilon$
Output: learned orthonormal dictionary $\mathbf{D}_K$, sparse representation matrix $\mathbf{X}_K$
Initialization : $\mathbf{P}_0 = \mathbf{0}$
for $k = 1$ to $K$ do
    Draw a mini-batch of data $\mathbf{Y}_{k-1} \triangleq [\mathbf{y}_1^{(k-1)}, \mathbf{y}_2^{(k-1)}, \ldots, \mathbf{y}_R^{(k-1)}]$ from a data source;
    Normalization: $\mathbf{y}_i^{(k-1)} \leftarrow \mathbf{y}_i^{(k-1)}/\|\mathbf{y}_i^{(k-1)}\|_2$, $\forall i = 1, \ldots, R$;
    $\mathbf{D} = \mathbf{D}_{k-1}$;
    while $\|\mathbf{Y}_{k-1} - \mathbf{D}\mathbf{X}_{k-1}\|_F^2 + \lambda\|\mathbf{X}_{k-1}\|_0$ not converged with error bound $\epsilon$ do
        $\mathbf{C}_{k-1} = \mathbf{D}^T\mathbf{Y}_{k-1}$;
        $[\mathbf{X}_{k-1}]_{ij} = \begin{cases} [\mathbf{C}_{k-1}]_{ij}\,, & \left|[\mathbf{C}_{k-1}]_{ij}\right| \geq \sqrt{\lambda} \\ 0, & \left|[\mathbf{C}_{k-1}]_{ij}\right| < \sqrt{\lambda} \end{cases}$;
        $\mathbf{P}_k = \mathbf{P}_{k-1} + \mathbf{X}_{k-1}\mathbf{Y}_{k-1}^T$;
        $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{P}_k$ ; // Compute SVD
        $\mathbf{D} = \mathbf{V}\mathbf{U}^T$;
    end
    $\mathbf{D}_k = \mathbf{D}$;
end
```

Algorithm 3.1: Online Orthonormal Dictionary Learning

into the range $[0, 1]$ to form the matrix $\mathbf{Y}_{k-1} \in \mathbb{R}^{N\times R}$. Then we use the previous dictionary $\mathbf{D}_{k-1} \in \mathbb{R}^{N\times N}$ as a warm start to represent $\mathbf{Y}_{k-1}$ with sparse coefficients $\mathbf{X}_{k-1} \in \mathbb{R}^{N\times R}$ by hard thresholding with $\sqrt{\lambda}$, and obtain the updated dictionary $\mathbf{D}_k$ for the following k-th FWI iteration with the orthonormal matrices of singular vectors of $\mathbf{P}_k \in \mathbb{R}^{N\times N}$ that accumulates $\mathbf{X}_i\mathbf{Y}_i^T$ for $i = 0, 1, \ldots, k-1$. Essentially, the above two alternating steps for learning $\mathbf{D}_k$ keep reducing the value of the function

$$\hat{e}_k(\mathbf{D}) \triangleq \frac{1}{kR}\sum_{i=0}^{k-1}\left(\|\mathbf{Y}_i - \mathbf{D}\mathbf{X}_i\|_F^2 + \lambda\|\mathbf{X}_i\|_0\right) \tag{3.24}$$

which, in effect, takes training patches of all previously optimized model perturbations $\{\delta\mathbf{m}_i\}_{i=0}^{k-1}$ into account. It is proved in [16] that $\hat{e}_k(\mathbf{D})$ converges to $e(\mathbf{D})$ with probability one if k is sufficiently large and, therefore, the online orthonormal dictionary learning converges to a stationary point.

3.4 Full Waveform Inversion with Dictionary-based Sparsity Regularization

Recall the LASSO optimization problem of each Gauss-Newton iteration $k = 0, 1, 2, \ldots$ under the CS framework:

$$\min_{\boldsymbol{\alpha}} \left\{ J_k^{(\mathrm{W})}(\boldsymbol{\alpha}) \triangleq \frac{1}{2} \|\mathbf{W}_k \delta \mathbf{d}_k - \mathbf{W}_k \mathbf{J}_k \boldsymbol{\mathcal{D}}_k(\boldsymbol{\alpha})\|_2^2 \right\} \quad \text{s.t.} \quad \|\boldsymbol{\alpha}\|_1 \leq \tau_k \quad \text{(3.12 revisited)}$$

where right now $\boldsymbol{\mathcal{D}}_k$ is the SOT synthesis operator based on the orthonormal dictionary $\mathbf{D}_k$ trained for the k-th FWI iteration and $\boldsymbol{\alpha}$ is the SOT coefficient vector. If $\boldsymbol{\alpha}_k$ minimizes the objective function $J_k^{(\mathrm{W})}(\boldsymbol{\alpha})$ in (3.12), then its inverse SOT recovers the optimal model perturbation $\delta \mathbf{m}_k$ via

$$\delta \mathbf{m}_k = \boldsymbol{\mathcal{D}}_k(\boldsymbol{\alpha}_k). \tag{3.25}$$

In (3.12) the subsampling matrix $\mathbf{W}_k$ must be designed. The construction of $\mathbf{W}_k$ can take advantage of the linearity property of wave equations. Because the computational cost of an FWI iteration is in proportion to the number of seismic wave modeling processes with respect to different source functions, a random source encoding method has been proposed to combine a large number of sequential sources with random weights into only a few simultaneous shots. These simultaneous shots are named *supershots* in the literature [8, 14, 64, 109, 123]. Due to the linear relationship between a seismic wavefield and its source function, the random weights can be incorporated into $\mathbf{W}_k$ and become the key to subsampling. Summing many individual sources into a few simultaneous shots introduces crosstalk artifacts. Nevertheless, crosstalk can be mitigated during the inversion process by enforcing a sparsity constraint in the SOT domain.

3.4.1 Random Source Encoding – The Supershot Method

Consider a conventional seismic survey with N_s shots, in which each shot produces a 2D acoustic seismic wavefield $p_j(\mathbf{x}, t; \mathbf{x}_{s_j})$ modeled by a time-domain PDE

$$\left(m(\mathbf{x})\frac{\partial^2}{\partial t^2} - \nabla^2\right) p_j(\mathbf{x}, t; \mathbf{x}_{s_j}) = f(t)\delta(\mathbf{x} - \mathbf{x}_{s_j}), \quad \forall j = 1, \ldots, N_s \tag{3.26}$$

where $f(t)\delta(\mathbf{x} - \mathbf{x}_{s_j})$ is the point source function excited at location $\mathbf{x}_{s_j}$.

The random source encoding method chooses a random time series $w_j(t)$ for each source such that each element of $w_j(t)$ is an i.i.d. $\mathcal{N}(0, 1)$ random variable. Based on the linearity of the Green's function, if the point source function is replaced by $(w_j(t) * f(t))\delta(\mathbf{x} - \mathbf{x}_{s_j})$, where $*$ denotes convolution in the time domain, then the output wavefield $q_j(\mathbf{x}, t; \mathbf{x}_{s_j})$ modeled by the wave equation

$$\left(m(\mathbf{x})\frac{\partial^2}{\partial t^2} - \nabla^2\right) q_j(\mathbf{x}, t; \mathbf{x}_{s_j}) = (w_j(t) * f(t))\delta(\mathbf{x} - \mathbf{x}_{s_j}), \quad \forall j = 1, \ldots, N_s \tag{3.27}$$

can be expressed as

$$q_j(\mathbf{x}, t; \mathbf{x}_{s_j}) = w_j(t) * p_j(\mathbf{x}, t; \mathbf{x}_{s_j}), \quad \forall j = 1, \ldots, N_s. \tag{3.28}$$

Convolving all source functions $f(t)\delta(\mathbf{x} - \mathbf{x}_{s_j})$ with different Gaussian time series $w_j(t)$ and then stacking them together generates an encoded simultaneous shot, which is also termed as a *supershot* in the literature. Because all $w_j(t)$ are stochastically independent, any number of stochastically independent supershots can be generated in this way by repeating the process. Suppose N'_s supershots are used for seismic modeling, $N'_s \ll N_s$, then each supershot is defined by

$$f_i^{(\mathrm{s})}(\mathbf{x}, t) = \sum_{j=1}^{N_s} (w_{ij}(t) * f(t))\delta(\mathbf{x} - \mathbf{x}_{s_j}), \quad \forall i = 1, \ldots, N'_s, \tag{3.29}$$

where i is the supershot index and $w_{ij}(t)$ is an independent random Gaussian time series that encodes $f(t)\delta(\mathbf{x} - \mathbf{x}_{s_j})$ for the i-th supershot. Similarly, the following wave equation that models the supershot wavefield $p_i^{(\mathrm{s})}(\mathbf{x}, t)$

$$\left(m(\mathbf{x})\frac{\partial^2}{\partial t^2} - \nabla^2\right) p_i^{(\mathrm{s})}(\mathbf{x}, t) = f_i^{(\mathrm{s})}(\mathbf{x}, t), \quad \forall i = 1, \ldots, N'_s \tag{3.30}$$

has the solution

$$p_i^{(\mathrm{s})}(\mathbf{x},t) = \sum_{j=1}^{N_s} w_{ij}(t) * p_j(\mathbf{x},t;\mathbf{x}_{s_j}), \quad \forall i = 1,\ldots,N_s'. \tag{3.31}$$

Modeling supershots and their corresponding wavefields in the frequency domain is a more common practice in recent research [8]. Since convolution in the time domain corresponds to multiplication in the frequency domain, a frequency-domain supershot at the frequency ω has the expression

$$\hat{f}_i^{(\mathrm{s})}(\mathbf{x};\omega) = \sum_{j=1}^{N_s} \hat{w}_{ij}(\omega)\hat{f}(\omega)\delta(\mathbf{x}-\mathbf{x}_{s_j}), \quad \forall i = 1,\ldots,N_s', \tag{3.32}$$

and the excited wavefield becomes

$$\hat{p}_i^{(\mathrm{s})}(\mathbf{x};\omega) = \sum_{j=1}^{N_s} \hat{w}_{ij}(\omega)\hat{p}_j(\mathbf{x};\omega,\mathbf{x}_{s_j}), \quad \forall i = 1,\ldots,N_s'. \tag{3.33}$$

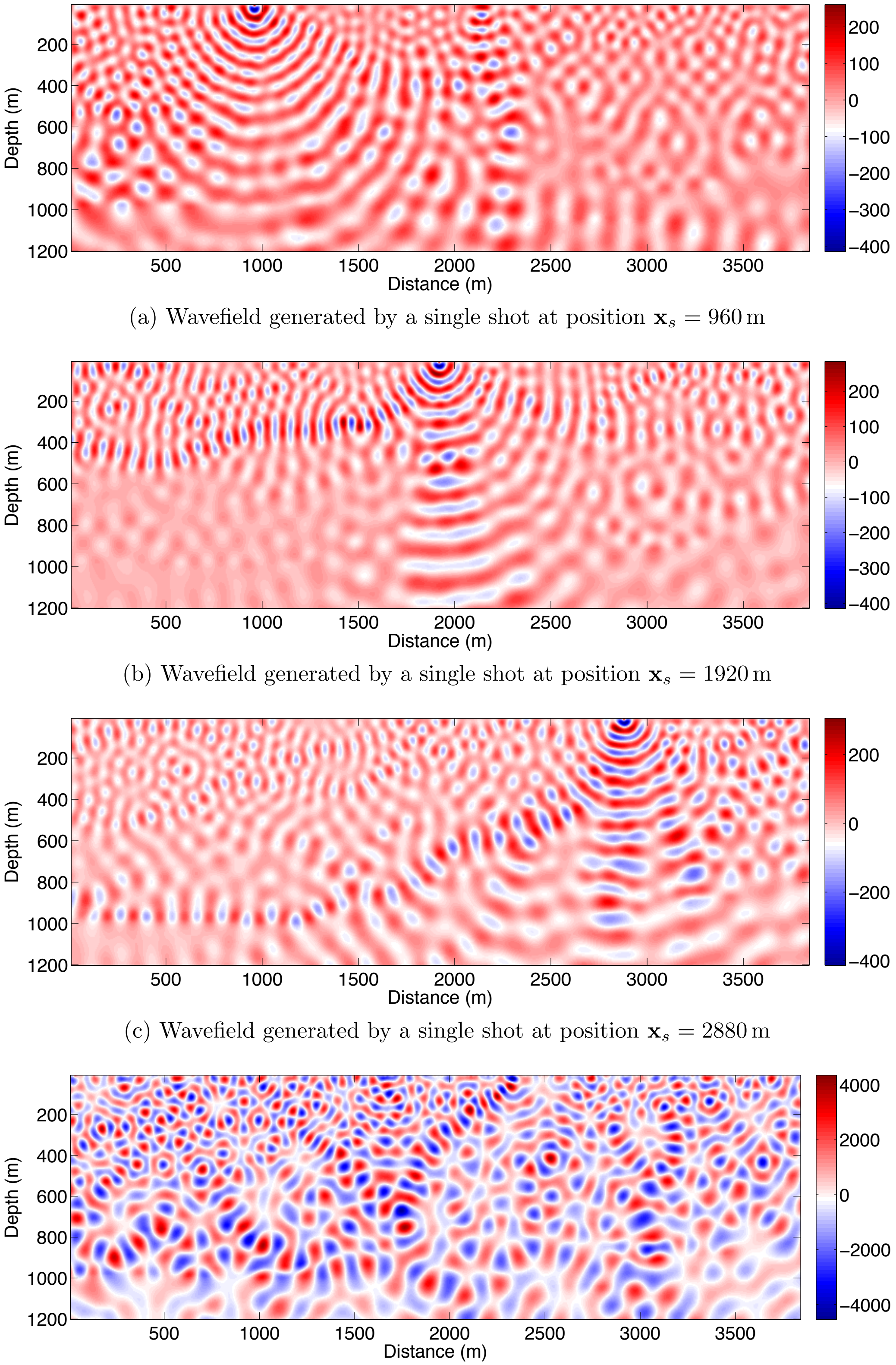
Figure 3.6 illustrates several frequency-domain wavefield examples with the frequency $\omega/(2\pi) = 22.8\,\mathrm{Hz}$, in which Figures 3.6(a), 3.6(b) and 3.6(c) show three regular wavefields $\hat{p}(\mathbf{x};\omega,\mathbf{x}_s)$ generated by three single shots at positions $\mathbf{x}_s = 960\,\mathrm{m}$, $1920\,\mathrm{m}$ and $2880\,\mathrm{m}$, respectively, and Figure 3.6(d) shows a supershot wavefield $\hat{p}_i^{(\mathrm{s})}(\mathbf{x};\omega)$ which encodes $N_s = 384$ shots on the surface with random Gaussian weights.

FWI uses the wavefield sample set $\mathbf{d}^{(\mathrm{s})} \triangleq \left\{\hat{p}_i^{(\mathrm{s})}(\mathbf{x}_r;\omega)\right\}$ collected at all receiver locations $\mathbf{x}_r$ for all supershots with different frequencies ω. Since each frequency is processed independently in frequency-domain modeling, the number of frequencies used in FWI can also be reduced to $N_\omega' < N_\omega$, and this set of frequencies can be randomly selected among all N_ω frequencies, which then reduces the dimension of $\mathbf{d}^{(\mathrm{s})}$ to $N_\omega' N_s' N_r$. According to (3.33), the relationship between $\mathbf{d}^{(\mathrm{s})}$ and the full-dimension data $\mathbf{d}$ for all receivers, single shots and frequencies can be written in a compact matrix form as

$$\mathbf{d}^{(\mathrm{s})} = \mathbf{W}\mathbf{d}. \tag{3.34}$$

The subsampling matrix $\mathbf{W}$ of size $N_\omega' N_s' N_r \times N_\omega N_s N_r$ is structured as

$$\mathbf{W} \triangleq \mathrm{diag}\left\{\hat{\mathbf{w}}(\omega_1),\ldots,\hat{\mathbf{w}}(\omega_{N_\omega'})\right\} \otimes \mathbf{I} \tag{3.35}$$



(a) Wavefield generated by a single shot at position $\mathbf{x}_s = 960\,\mathrm{m}$

(b) Wavefield generated by a single shot at position $\mathbf{x}_s = 1920\,\mathrm{m}$

(c) Wavefield generated by a single shot at position $\mathbf{x}_s = 2880\,\mathrm{m}$

(d) Wavefield generated by a supershot encoding $N_s = 384$ shots with random Gaussian weights

Figure 3.6: Wavefield examples generated by a single shot and a supershot with frequency 22.8 Hz

where each $\hat{\mathbf{w}}(\omega)$ is a random matrix of size $N_s' \times N_s$ whose (i, j)-th entry is $\hat{w}_{ij}(\omega)$, the different $\hat{\mathbf{w}}(\omega)$ for the selected N_ω' frequencies are assembled together as a block diagonal matrix, and the operator $\otimes$ denotes the Kronecker product whose right operand $\mathbf{I}$ is an identity matrix of size $N_r \times N_r$.

According to the perturbation analysis based on Born approximation theory, the wavefield perturbation $\widehat{\delta p}^{(\mathrm{s})}(\mathbf{x}; \omega)$ on the background wavefield $\hat{p}^{(\mathrm{s})}(\mathbf{x}; \omega)$ attributed to a small model perturbation $\delta\mathbf{m}$ on the background model $\mathbf{m}$ satisfies the equation

$$\left(-m(\mathbf{x})\omega^2 - \nabla^2\right) \widehat{\delta p}^{(\mathrm{s})}(\mathbf{x}; \omega) = \omega^2 \delta m(\mathbf{x}) \hat{p}^{(\mathrm{s})}(\mathbf{x}; \omega). \tag{3.36}$$

Similar to the regular point source case in (3.8), $\widehat{\delta p}^{(\mathrm{s})}(\mathbf{x}_r; \omega)$ can be expressed as

$$\widehat{\delta p}^{(\mathrm{s})}(\mathbf{x}_r; \omega) = \omega^2 \hat{f}(\omega) \sum_{\mathbf{x} \in \mathcal{U}} \delta m(\mathbf{x}) \hat{G}(\mathbf{x}_r; \omega, \mathbf{x}) \sum_{j=1}^{N_s} \hat{w}_j(\omega) \hat{G}(\mathbf{x}; \omega, \mathbf{x}_{s_j}), \tag{3.37}$$

yielding the (i, j)-th entry of the Jacobian sub-matrix $\mathbf{J}^{(\mathrm{s})}(\omega)$ as

$$\begin{aligned} \left[\mathbf{J}^{(\mathrm{s})}(\omega)\right]_{ij} &\triangleq \lim_{\delta m(\mathbf{x}_j) \to 0} \frac{\widehat{\delta p}^{(\mathrm{s})}(\mathbf{x}_{r_i}; \omega)}{\delta m(\mathbf{x}_j)} \\ &= \omega^2 \hat{f}(\omega) \hat{G}(\mathbf{x}_{r_i}; \omega, \mathbf{x}_j) \sum_{l=1}^{N_s} \hat{w}_l(\omega) \hat{G}(\mathbf{x}_j; \omega, \mathbf{x}_{s_l}). \end{aligned} \tag{3.38}$$

The entire Jacobian matrix $\mathbf{J}^{(\mathrm{s})}$ stacks $\mathbf{J}^{(\mathrm{s})}(\omega)$ for all supershots and all frequencies together, and its relationship between the full-dimension Jacobian matrix $\mathbf{J}$ for all receivers, single shots and frequencies can be written as

$$\mathbf{J}^{(\mathrm{s})} = \mathbf{W}\mathbf{J}. \tag{3.39}$$

For each FWI iteration k, all $\hat{w}_{ij}(\omega)$ can be regenerated so that the random subsampling matrix varies with the iterations and can be denoted as $\mathbf{W}_k$. This approach suppresses crosstalk artifacts into incoherent Gaussian noise and yields better reconstruction results. Meanwhile, no artificial bias towards a specific random source encoding pattern would be introduced into the solution by redrawing the random subsampling matrix. Such an approach has been recommended in previous research on CS [62, 67, 79, 88] and FWI [56, 57, 73, 137].

Therefore, in (3.12), $\mathbf{W}_k \delta \mathbf{d}_k$ can be obtained as a whole by calculating the difference between the recorded receiver data $\mathbf{d}_{\text{obs}}^{(\text{s})} \triangleq \mathbf{W}_k \mathbf{d}_{\text{obs}}$ encoded by $\mathbf{W}_k$ and the calculated receiver data $\mathbf{d}_k^{(\text{s})}$ generated by supershots. $\mathbf{W}_k \mathbf{J}_k$ can be regarded as the compressive Jacobian whose components includes non-altered Green's functions for receivers and random encoded Green's functions for sources.

The solution $\boldsymbol{\alpha}_k$ of the LASSO problem (3.12) relies on the choice of the sparsity constraint τ_k. As suggested by [133], every LASSO problem implies a convex and non-increasing function $\phi(\tau)$ that associates the least-squares residual to the sparsity level τ. In this problem, each FWI iteration $k = 0, 1, 2, \ldots$ needs to solve (3.12) and, therefore, has an implicit $\phi_k(\tau)$. Following the same idea used by [57, 73, 74], one can estimate τ_k by using a linear approximation of $\phi_k'(\tau)$ at $\tau = 0$, given in Theorem 2.1 of [133]

$$\tau_k \approx -\frac{\phi_k(0)}{\phi_k'(0)} = \frac{\|\mathbf{W}_k \delta \mathbf{d}_k\|_2^2}{\left\| \boldsymbol{\mathcal{D}}_k^{\dagger} \left([\mathbf{W}_k \mathbf{J}_k]^{\dagger} (\mathbf{W}_k \delta \mathbf{d}_k) \right) \right\|_{\infty}} \tag{3.40}$$

where $\|\cdot\|_\infty$ is the maximum norm, $[\mathbf{W}_k \mathbf{J}_k]^\dagger$ is the adjoint of the compressive Jacobian $\mathbf{W}_k \mathbf{J}_k$ and performs the following calculation over the vector $\mathbf{W}_k \delta \mathbf{d}_k$

$$\begin{aligned} &[\mathbf{W}_k \mathbf{J}_k]^{\dagger} (\mathbf{W}_k \delta \mathbf{d}_k) \\ =& \sum_{m=1}^{N_\omega'} \omega_m^2 \hat{f}(\omega_m) \sum_{n=1}^{N_r} G_k(\mathbf{x}_{r_n}; \omega_m, \mathbf{x}) \sum_{i=1}^{N_s'} \sum_{j=1}^{N_s} [\hat{\mathbf{w}}_k(\omega_m)]_{ij} \, G_k(\mathbf{x}; \omega_m, \mathbf{x}_{s_j}) \left[\delta p_k^{(\text{s})} \right]_i (\mathbf{x}_{r_n}; \omega_m). \end{aligned} \tag{3.41}$$

The above expression is a function with respect to the medium grid points $\mathbf{x}$, so that it can be interpreted as a wavefield or image and hence can also be decomposed into global SOT coefficient by $\boldsymbol{\mathcal{D}}_k^\dagger$.

3.4.2 Projected Quasi-Newton Method for solving the LASSO problems

The computational complexity of the FWI problem is reduced considerably after reducing the data dimensionality from $N_\omega N_s N_r$ to $N'_\omega N'_s N_r$. However, in order to minimize the objective function $J_k^{(\mathrm{W})}(\boldsymbol{\alpha})$ in (3.12) with sparsity promotion on the global SOT coefficient $\boldsymbol{\alpha}$, the descent direction of $\boldsymbol{\alpha}$ must be projected into an ℓ_1-norm ball with radius τ_k.

```
Input: Step length bounds $0 < a_{\min} < a_{\max}$, $\boldsymbol{\gamma}^{(l)}$, $\mathbf{B}^{(l)}$, sufficient decrease
       parameter $\nu$, $\ell_1$-norm bound $\tau_k$
Initialization: $\boldsymbol{\alpha}_{\text{new}} \leftarrow \boldsymbol{\alpha}^{(l)}$, initial step length $a = \frac{\left[\mathbf{p}^{(l-1)}\right]^T \mathbf{p}^{(l-1)}}{\left[\mathbf{p}^{(l-1)}\right]^T \mathbf{q}^{(l-1)}}$, function
                maximum $f_{\max} \leftarrow -\infty$
while not converged do
    $\boldsymbol{\alpha}_{\text{old}} \leftarrow \boldsymbol{\alpha}_{\text{new}}$;
    $a \leftarrow \min\{a_{\max}, \max\{a_{\min}, a\}\}$;
    $\mathbf{d} \leftarrow \boldsymbol{\mathcal{P}}_{\tau_k}^{(\ell_1)}\left(\boldsymbol{\alpha}_{\text{old}} - a\nabla Q^{(l)}(\boldsymbol{\alpha}_{\text{old}})\right) - \boldsymbol{\alpha}_{\text{old}}$ ; // $\nabla Q^{(l)}(\boldsymbol{\alpha}) = \mathbf{B}^{(l)}\boldsymbol{\alpha} + \boldsymbol{\gamma}^{(l)}$
    $a \leftarrow 1$;
    $f_{\max} \leftarrow \max\left\{f_{\max}, J_k^{(W)}(\boldsymbol{\alpha}_{\text{old}})\right\}$;
    while $Q^{(l)}(\boldsymbol{\alpha}_{old} + a\mathbf{d}) > f_{\max} + \nu a \nabla Q^{(l)}(\boldsymbol{\alpha}_{old})^T\mathbf{d}$ do
        Choose $a \in (0, a)$ by backtracking;
    end
    $\boldsymbol{\alpha}_{\text{new}} \leftarrow \boldsymbol{\alpha}_{\text{old}} + a\mathbf{d}$;
    $a \leftarrow \frac{\left[\boldsymbol{\alpha}_{\text{new}} - \boldsymbol{\alpha}_{\text{old}}\right]^T \left[\boldsymbol{\alpha}_{\text{new}} - \boldsymbol{\alpha}_{\text{old}}\right]}{\left[\boldsymbol{\alpha}_{\text{new}} - \boldsymbol{\alpha}_{\text{old}}\right]^T \left[\nabla Q^{(l)}(\boldsymbol{\alpha}_{\text{new}}) - \nabla Q^{(l)}(\boldsymbol{\alpha}_{\text{old}})\right]}$
end
Output: $\widehat{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}_{\text{new}}$
```

Algorithm 3.2: Spectral Projected Gradient (SPG) Algorithm

A limited-memory projected quasi-Newton method (l-PQN) proposed by [112] can solve the LASSO problem (3.12) iteratively, based on a two-layer strategy. In each iteration $l = 0, 1, 2, \ldots$, the outer layer formulates a quadratic approximation function $Q^{(l)}(\boldsymbol{\alpha})$ of the objective function $J_k^{(\mathrm{W})}(\boldsymbol{\alpha})$ around the current iterate $\boldsymbol{\alpha}^{(l)}$

$$Q^{(l)}(\boldsymbol{\alpha}) \triangleq J_k^{(\mathrm{W})}\left(\boldsymbol{\alpha}^{(l)}\right) + \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(l)}\right)^T \boldsymbol{\gamma}^{(l)} + \frac{1}{2}\left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(l)}\right)^T \mathbf{B}^{(l)} \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(l)}\right) \qquad (3.42)$$

where $\boldsymbol{\gamma}^{(l)}$ is the gradient of $J_k^{(\mathrm{W})}(\boldsymbol{\alpha})$ evaluated for $\boldsymbol{\alpha}^{(l)}$

$$\boldsymbol{\gamma}^{(l)} \triangleq \frac{\partial J_k^{(\mathrm{W})}}{\partial \boldsymbol{\alpha}}\left(\boldsymbol{\alpha}^{(l)}\right) = -\Re\left\{\boldsymbol{\mathcal{D}}_k^{\dagger}\left([\mathbf{W}_k\mathbf{J}_k]^{\dagger}\left(\mathbf{W}_k\delta\mathbf{d}_k - \mathbf{W}_k\mathbf{J}_k\boldsymbol{\mathcal{D}}_k\left(\boldsymbol{\alpha}^{(l)}\right)\right)\right)\right\} \tag{3.43}$$

and $\mathbf{B}^{(l)}$ denotes a positive-definite approximation matrix of $[\mathbf{W}_k\mathbf{J}_k\boldsymbol{\mathcal{D}}_k]^{\dagger}\,\mathbf{W}_k\mathbf{J}_k\boldsymbol{\mathcal{D}}_k$, the Hessian matrix of $J_k^{(\mathrm{W})}(\boldsymbol{\alpha})$, at the l-th iteration of l-PQN. The inner layer iteratively searches for a feasible descent direction by minimizing $Q^{(l)}(\boldsymbol{\alpha})$ subject to the ℓ_1-norm constraints

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, Q^{(l)}(\boldsymbol{\alpha}) \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_1 \leq \tau_k. \tag{3.44}$$

This problem can be solved via the spectral projected gradient (SPG) algorithm [12, 13] shown in Algorithm 3.2. For the sake of convenience, the following variables are defined

$$\begin{aligned}\mathbf{p}^{(l)} &\triangleq \boldsymbol{\alpha}^{(l+1)} - \boldsymbol{\alpha}^{(l)} \\ \mathbf{q}^{(l)} &\triangleq \boldsymbol{\gamma}^{(l+1)} - \boldsymbol{\gamma}^{(l)}.\end{aligned} \tag{3.45}$$

In Algorithm 3.2, the Euclidean projection operator $\boldsymbol{\mathcal{P}}_{\tau}^{(\ell_1)}(\boldsymbol{\alpha})$ that projects the vector $\boldsymbol{\alpha}$ onto the ℓ_1-norm ball with radius τ is defined as

$$\boldsymbol{\mathcal{P}}_{\tau}^{(\ell_1)}(\boldsymbol{\alpha}) \triangleq \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, \|\boldsymbol{\alpha} - \boldsymbol{\beta}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_1 = \tau. \tag{3.46}$$

A randomized algorithm that efficiently solves this projection problem is shown in Algorithm 3.3 [42].

```
Input: $\boldsymbol{\alpha} \in \mathbb{R}^N$, $\tau > 0$
Sort $\boldsymbol{\alpha}$ s.t. $|\alpha_1| \geq |\alpha_2| \geq \cdots \geq |\alpha_N|$;
Find $\rho = \underset{j}{\operatorname{argmax}}\left(|\alpha_j| - \frac{1}{j}\left(\sum_{r=1}^{j}|\alpha_r| - \tau\right)\right)$;
Define $\theta = \frac{1}{\rho}\left(\sum_{i=1}^{\rho}|\alpha_i| - \tau\right)$;
Output: $\boldsymbol{\beta} \in \mathbb{R}^N$ such that $\beta_i = \operatorname{sign}(\alpha_i) \cdot \max\{|\alpha_i| - \theta, 0\}$
```

Algorithm 3.3: Projection onto an ℓ_1-norm ball

After solving the inner-layer problem (3.44), the direction $\mathbf{d}^{(l)} \triangleq \widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^{(l)}$ is guaranteed to be a feasible descent direction since $\mathbf{B}^{(l)}$ is a positive-definite matrix.

In order to obtain the next iterate $\boldsymbol{\alpha}^{(l+1)}$ for the outer layer, a backtracking line search method along the search direction $\mathbf{d}^{(l)}$ can be applied to find a step length $a \in (0, 1]$ such that the Armijo condition [5]

$$J_k^{(\mathrm{W})}\left(\boldsymbol{\alpha}^{(l)} + a\mathbf{d}^{(l)}\right) \leq J_k^{(\mathrm{W})}\left(\boldsymbol{\alpha}^{(l)}\right) + \nu a \left[\boldsymbol{\gamma}^{(l)}\right]^T \mathbf{d}^{(l)} \tag{3.47}$$

which ensures a sufficient decrease on the objective function is satisfied. In (3.47) the sufficient decrease parameter ν is set to 10^{-4} as suggested by Nocedal [94]. Because $\mathbf{d}^{(l)}$ takes the ℓ_1-norm constraint into account, the next iterate $\boldsymbol{\alpha}^{(l+1)}$ also satisfies the constraint for the selected value of a.

The positive-definite matrix $\mathbf{B}^{(l)}$ that approximates the Hessian matrix of $J_k^{(\mathrm{W})}(\boldsymbol{\alpha})$ can be built with the quasi-Newton methods, among which the limited-memory Broyden-Fletcher-Goldfarb-Shanno (l-BFGS) algorithm [93] is one of the most popular members. The l-BFGS algorithm maintains at most m past $\mathbf{p}^{(l)}$ and $\mathbf{q}^{(l)}$ vectors for the Hessian approximation. It initializes $\mathbf{B}^{(0)} = \sigma^{(0)}\mathbf{I}$, and for $l > 0$, updates $\mathbf{B}^{(l)}$ by the following formula

$$\mathbf{B}^{(l)} = \sigma^{(l)}\mathbf{I} - \begin{bmatrix} \sigma^{(l)}\mathbf{P}^{(l)} & \mathbf{Q}^{(l)} \end{bmatrix} \begin{bmatrix} \sigma^{(l)}\left[\mathbf{P}^{(l)}\right]^T \mathbf{P}^{(l)} & \mathbf{L}^{(l)} \\ \left[\mathbf{L}^{(l)}\right]^T & -\mathbf{X}^{(l)} \end{bmatrix}^{-1} \begin{bmatrix} \sigma^{(l)}\left[\mathbf{P}^{(l)}\right]^T \\ \left[\mathbf{Q}^{(l)}\right]^T \end{bmatrix} \tag{3.48}$$

where the scalar $\sigma^{(l)} \triangleq \dfrac{\left[\mathbf{q}^{(l)}\right]^T \mathbf{p}^{(l)}}{\left[\mathbf{q}^{(l)}\right]^T \mathbf{q}^{(l)}}$, the matrices $\mathbf{P}^{(l)} \triangleq \left[\mathbf{p}^{(l-m)}, \ldots, \mathbf{p}^{(l-1)}\right]$, $\mathbf{Q}^{(l)} \triangleq \left[\mathbf{q}^{(l-m)}, \ldots, \mathbf{q}^{(l-1)}\right]$, $\mathbf{X}^{(l)} \triangleq \operatorname{diag}\left\{\left[\mathbf{p}^{(l-m)}\right]^T \mathbf{q}^{(l-m)}, \ldots, \left[\mathbf{p}^{(l-1)}\right]^T \mathbf{q}^{(l-1)}\right\}$ and $\mathbf{L}^{(l)}$ is defined by

$$\left[\mathbf{L}^{(l)}\right]_{ij} = \begin{cases} \left[\mathbf{p}^{(l-m-1+i)}\right]^T \mathbf{q}^{(l-m-1+j)}, & \text{if} \quad i > j \\ 0, & \text{otherwise.} \end{cases}$$

Finally, Algorithm 3.4 summarizes the overall SOT-based sparse-promoting FWI optimization procedure which is initialized by a smooth model $\mathbf{m}_{\mathrm{smth}}$. The accuracy of $\mathbf{m}_{\mathrm{smth}}$ directly affects the performance of FWI. To avoid FWI becoming trapped in local minima, a good initial model can be found using other inversion methods such as traveltime tomography [17] or migration velocity analysis [124]. Each newly optimized $\delta\mathbf{m}_k$ becomes the source for R new patches for online dictionary learning in order to update the dictionary to $\mathbf{D}_{k+1}$, which will then be used in the corresponding SOT operator $\boldsymbol{\mathcal{D}}_{k+1}$ for the sparse representation of $\delta\mathbf{m}_{k+1}$ in the next FWI iteration. The entire workflow of the compressive FWI using the SOT is depicted in Figure 3.7.

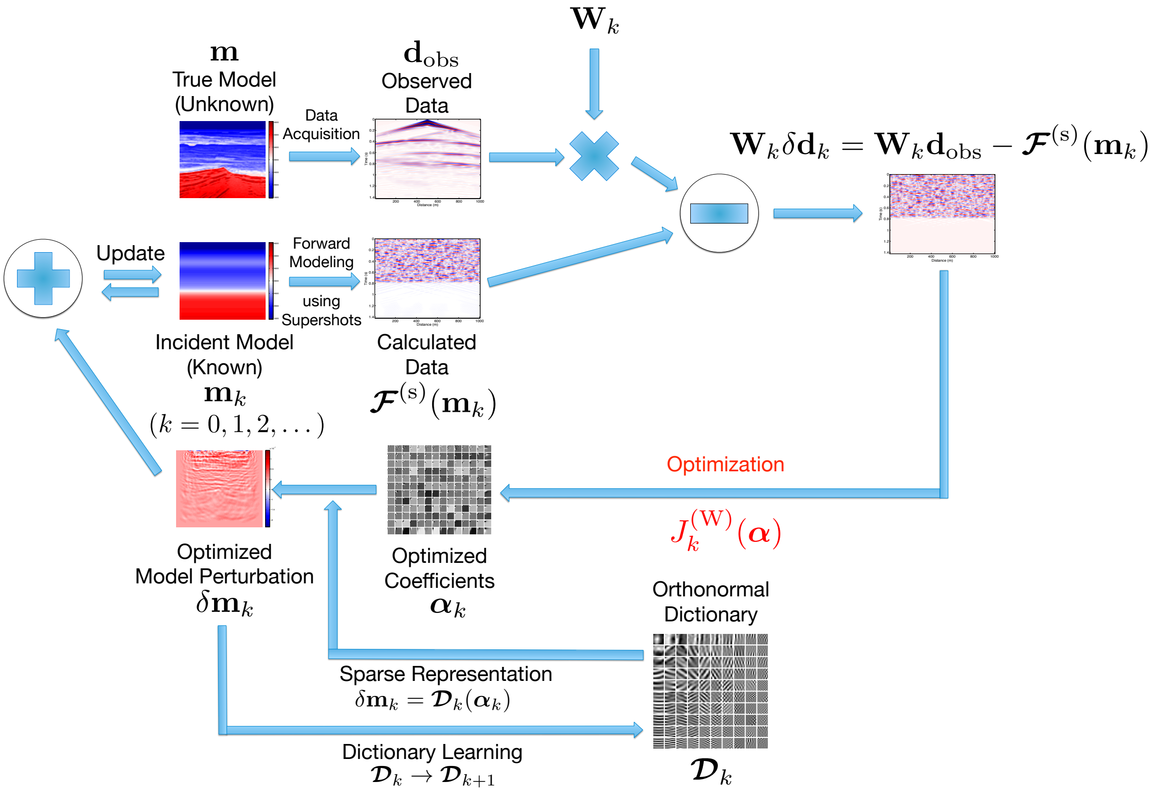

$\mathbf{W}_k$
$\mathbf{m}$
True Model (Unknown)
Data Acquisition
$\mathbf{d}_{\text{obs}}$
Observed Data
$\mathbf{W}_k\delta\mathbf{d}_k = \mathbf{W}_k\mathbf{d}_{\text{obs}} - \boldsymbol{\mathcal{F}}^{(\text{s})}(\mathbf{m}_k)$
Update
Forward Modeling
using Supershots
Incident Model (Known)
$\mathbf{m}_k$
$(k = 0, 1, 2, \dots)$
Calculated Data
$\boldsymbol{\mathcal{F}}^{(\text{s})}(\mathbf{m}_k)$
Optimization
$J_k^{(\text{W})}(\boldsymbol{\alpha})$
Optimized Model Perturbation
$\delta\mathbf{m}_k$
Optimized Coefficients
$\boldsymbol{\alpha}_k$
Orthonormal Dictionary
Sparse Representation
$\delta\mathbf{m}_k = \boldsymbol{\mathcal{D}}_k(\boldsymbol{\alpha}_k)$
Dictionary Learning
$\boldsymbol{\mathcal{D}}_k \rightarrow \boldsymbol{\mathcal{D}}_{k+1}$
$\boldsymbol{\mathcal{D}}_k$

Figure 3.7: FWI workflow using SOT-domain sparsity promotion with adaptive transform $\boldsymbol{\mathcal{D}}_k$ based on online orthonormal dictionary learning

3.5 Numerical Experiments

In the following experiments, the Gauss-Newton FWI is performed on full data using sequential point sources and compressive data using supershots. For the compressive FWI, two kinds of transforms are used to promote the sparsity of the model perturbation $\delta\mathbf{m}$ in transform domains, in which one is the fixed and non-adaptive

```
Input: Recorded seismic data $\mathbf{d}_{\text{obs}} \triangleq \{\hat{p}_{\text{obs}}(\mathbf{x}_r;\omega,\mathbf{x}_s)\}$, initial smooth model
       $\mathbf{m}_{\text{smth}}$, number of FWI iterations $K$, receiver locations $\mathbf{x}_r \in \mathcal{S}$, number
       of receivers $N_r$, sequential shot locations $\mathbf{x}_s \in \mathcal{S}$, number of sequential
       shots $N_s$, number of supershots $N_s'$, number of frequencies $N_\omega$, reduced
       number of frequencies $N_\omega'$, patch height $n_z$, patch width $n_x$, atom size
       $N = n_z n_x$, convergence error bound $\epsilon$
Output: FWI result $\mathbf{m}_K$
Initialization: $k \leftarrow 0$, $\mathbf{m}_0 \leftarrow \mathbf{m}_{\text{smth}}$, relative model change $\Delta_0 \leftarrow \infty$
while $\Delta_k > \epsilon$ and $k < K$ do
    Randomly draw $N_\omega'$ out of $N_\omega$ frequencies to form a set $\Omega'$;
    Generate $N_\omega'$ random Gaussian matrices $\hat{\mathbf{w}}_k(\omega) \triangleq \{\hat{w}_{ij}(\omega)\} \in \mathbb{R}^{N_s' \times N_s}$ for all
    frequencies $\omega \in \Omega'$ to produce $\mathbf{W}_k \triangleq \operatorname{diag}\left\{\hat{\mathbf{w}}_k(\omega_1), \ldots, \hat{\mathbf{w}}_k(\omega_{N_\omega'})\right\} \otimes \mathbf{I}$;
    Generate supershots $\hat{f}_i^{(\text{s})}(\mathbf{x};\omega) = \sum_{j=1}^{N_s} \hat{w}_{ij}(\omega)\hat{f}(\omega)\delta(\mathbf{x}-\mathbf{x}_{s_j}),\ \forall i = 1, \ldots, N_s'$;
    Encode the recorded seismic data $\mathbf{d}_{\text{obs}}^{(\text{s})} \triangleq \mathbf{W}_k \mathbf{d}_{\text{obs}}$;
    Solve (3.30) to get $\hat{p}_i^{(\text{s})}(\mathbf{x};\omega)$ for all supershots $\forall i = 1, \ldots, N_s'$, and
    frequencies $\omega \in \Omega'$;
    Collect $\mathbf{d}_k^{(\text{s})} \triangleq \left\{\hat{p}_i^{(\text{s})}(\mathbf{x}_r;\omega)\right\}$ for all receivers $\mathbf{x}_r \in \mathcal{S}$, supershots
    $\forall i = 1, \ldots, N_s'$, and frequencies $\omega \in \Omega'$;
    $\mathbf{W}_k \delta\mathbf{d}_k = \mathbf{d}_{\text{obs}}^{(\text{s})} - \mathbf{d}_k^{(\text{s})}$;
    Collect Green's functions $G_i^{(\text{s})}(\mathbf{x};\omega) \triangleq \sum_{j=1}^{N_s} \hat{w}_{ij}(\omega)\hat{G}(\mathbf{x};\omega,\mathbf{x}_{s_j})$ for all
    supershots $\forall i = 1, \ldots, N_s'$, and frequencies $\omega \in \Omega'$;
    Collect Green's functions $\hat{G}(\mathbf{x}_r;\omega,\mathbf{x}_j)$ for all receivers $\mathbf{x}_r \in \mathcal{S}$;
    Solve $\left\{\begin{array}{l} \boldsymbol{\alpha}_k = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{W}_k\delta\mathbf{d}_k - \mathbf{W}_k\mathbf{J}_k\boldsymbol{\mathcal{D}}_k(\boldsymbol{\alpha})\right\|_2^2 \\ \text{s.t. } \|\boldsymbol{\alpha}\|_1 \le \tau_k \approx \dfrac{\|\mathbf{W}_k\delta\mathbf{d}_k\|_2^2}{\left\|\boldsymbol{\mathcal{D}}_k^\dagger\left([\mathbf{W}_k\mathbf{J}_k]^\dagger(\mathbf{W}_k\delta\mathbf{d}_k)\right)\right\|_\infty} \end{array}\right\}$ with l-PQN;
    $\delta\mathbf{m}_k = \boldsymbol{\mathcal{D}}_k(\boldsymbol{\alpha}_k)$;
    Learn $\mathbf{D}_{k+1}$ from $R$ patches of $\delta\mathbf{m}_k$ using Algorithm 3.1 inside the outer
    for loop;
    $\mathbf{m}_{k+1} = \mathbf{m}_k + \delta\mathbf{m}_k$;
    $\Delta_k = \|\mathbf{m}_{k+1} - \mathbf{m}_k\|_2 / \|\mathbf{m}_k\|_2$;
    $k \leftarrow k + 1$;
end
```

Algorithm 3.4: Sparsity-Promoting FWI based on the SOT

curvelet transform and the other one is the proposed adaptive SOT.

Two benchmark velocity models are used to verify the inversion algorithms in realistic settings. First is the BG-Compass model whose exact form is shown in Figure 3.8(a). This model is rescaled to $N_z \times N_x = 100 \times 350$ grid points and covers a width of 3.5 km and a depth of 1 km. Full data generated by $N_s = 350$ shots are recorded by $N_r = 350$ receivers equispaced along the surface of the model. The well-known Marmousi model shown in Figure 3.9(a) serves as the second benchmark velocity model. This model is rescaled to $N_z \times N_x = 120 \times 384$ grid points and covers a width of 3.84 km and a depth of 1.2 km. Full data from $N_s = 384$ equispaced shots on the model surface are recorded over $N_r = 384$ equispaced receivers. Therefore, the grid spacing $\Delta x = \Delta z = 10\,\mathrm{m}$ guarantees that a sufficient number of grid points are used to represent the expected wavelengths and no grid dispersion happens. The wavefields are simulated by discretizing the PDE (3.30) with an 8th-order staggered-grid FDFD method [2] in which the left, right and bottom boundary reflections are absorbed by perfectly matched layers [63].

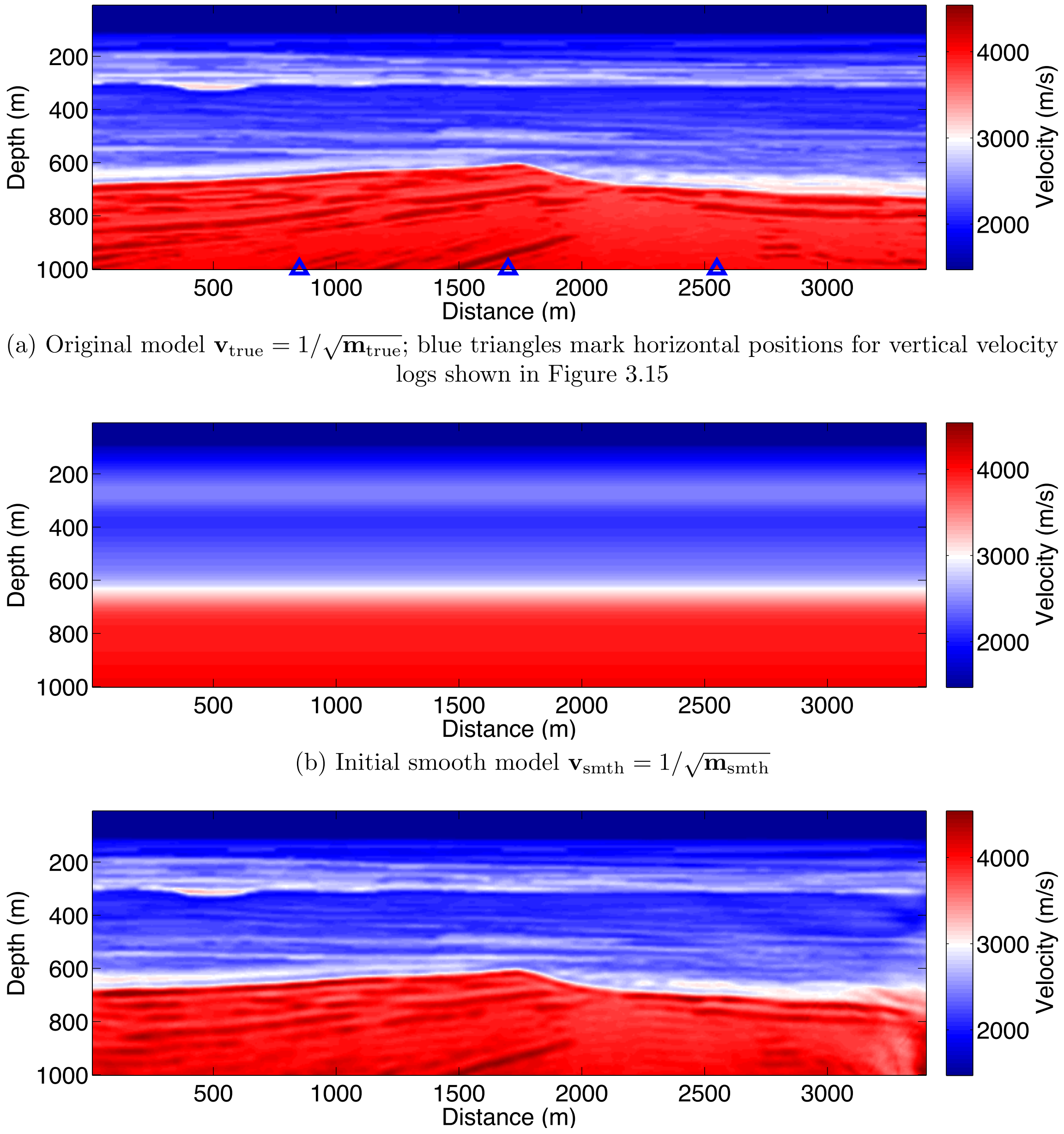
The shot source is a Ricker wavelet centered at 20 Hz with 256 frequency components spanning 3.0 to 48.1 Hz, and its spectrum $\hat{f}(\omega)$ is assumed known and fixed. FWI starts from an initial smooth model shown in Figure 3.8(b) for BG-Compass, or Figure 3.9(b) for Marmousi. In practical implementations, FWI is carried out in several consecutive frequency bands from low to high in order to avoid local minima caused by cycle skipping [18, 121]. Here FWI are performed across five frequency bands within the interval of 3.0 to 48.1 Hz, and thus the average number of frequencies per band is $N_\omega = 256/5 \approx 52$. In each frequency band, $K = 20$ FWI iterations are executed. After 20 FWI iterations are completed on one frequency band, the resulting more accurate model serves as the initial model for another 20 FWI iterations on the next higher frequency band. Although it is computationally expensive to perform FWI with the full data set from all N_s sequential shots and N_ω frequencies, these results were obtained and are shown in Figure 3.8(c) and Figure 3.9(c) for both models after 100 iterations.

For every FWI iteration using compressive data, only $N'_s = 3$ supershots and $N'_\omega = 16$ random frequencies from each frequency band are used. Thus, the problem dimensionality of the compressed objective function $J_k^{(\mathrm{W})}(\boldsymbol{\alpha})$ in (3.12) is $(N_\omega N_s)/(N'_\omega N'_s) \approx 400$ times smaller than that of the full-data Gauss-Newton objective function $J_k(\delta\mathbf{m})$ in (3.4). This does not necessarily mean that a compressive FWI iteration runs 400 times faster than a full-data FWI iteration as actual implementations may vary, but the reduced time on both forward modeling and objective function minimization, as well as the reduced memory costs, are still considerable.

3.5.1 Sparsity Regularization using Curvelets

Before the invention of the SOT using online orthonormal dictionary learning, the curvelet transform was the state-of-the-art method to exploit the sparsity of $\delta\mathbf{m}$ in FWI [57, 73, 74]. Figure 3.10 shows the workflow of compressive FWI using the curvelet transform for sparsity promotion. Each FWI iteration minimizes the objective function $J_k^{(\mathrm{W})}(\boldsymbol{\beta})$ defined in (3.12) where $\boldsymbol{\mathcal{D}}$ is specified as the fixed and non-adaptive curvelet transform $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\beta}$ has the curvelet coefficients for $\delta\mathbf{m}$.

The discrete curvelet transform based on the wrapping and mirror-extended techniques, which is implemented in the software *CurveLab* by [20, 33], is used here. The number of scales is set to 5, including the coarsest wavelet scale for the curvelet transform, and the number of angles from the second coarsest scale to the finest scale (the 5th scale) are 32, 64, 64, and 128, respectively. Such a curvelet transform has a complexity of $\mathcal{O}(n^2 \log n)$ for a model of size $n \times n$ [20]. Since no machine learning process is involved in the curvelet transform, its computational overhead is negligible compared to the forward modeling and can therefore be ignored. The inverted image results for the BG-Compass and Marmousi models using the curvelet-based method

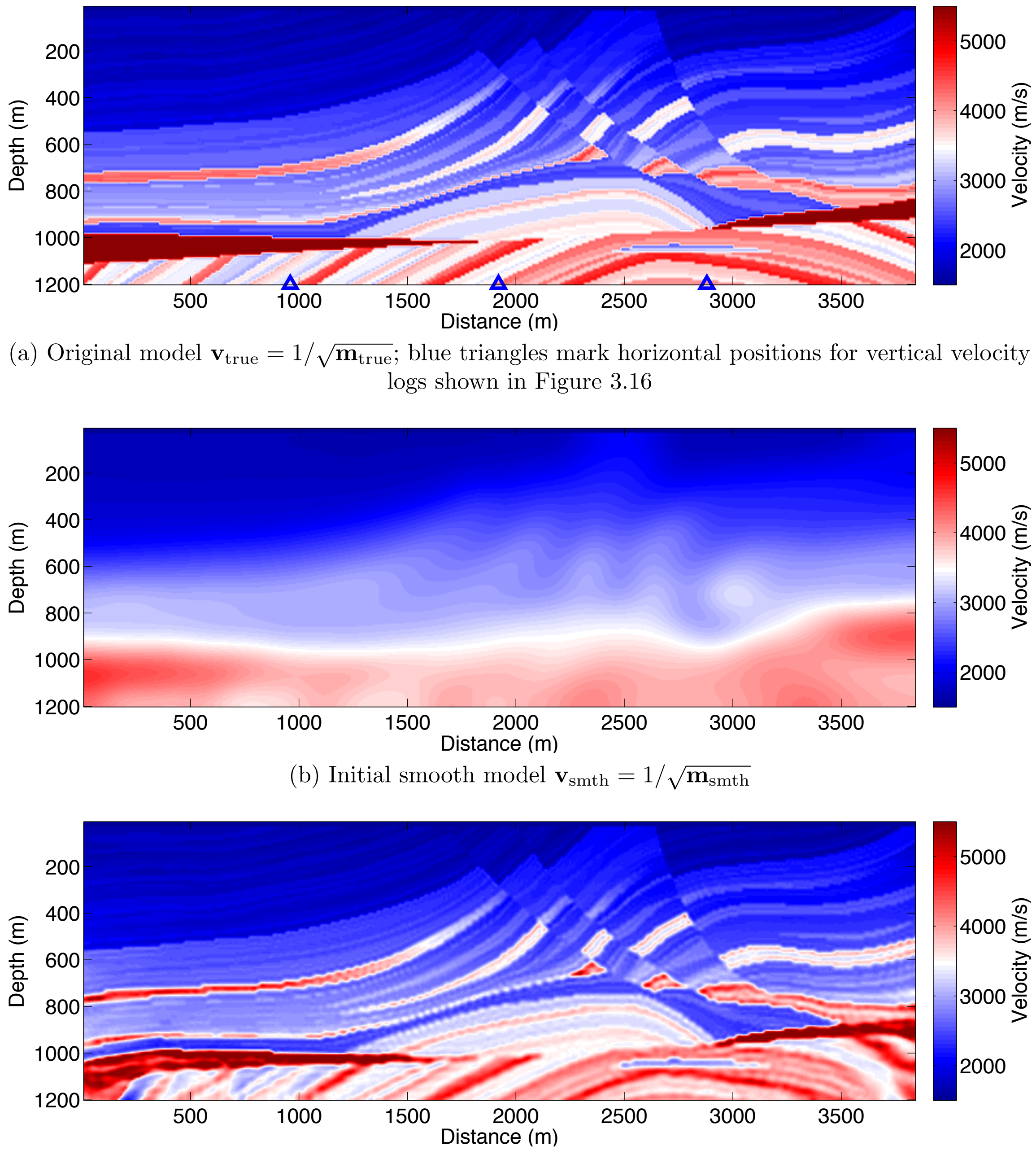


(a) Original model $\mathbf{v}_{\mathrm{true}} = 1/\sqrt{\mathbf{m}_{\mathrm{true}}}$; blue triangles mark horizontal positions for vertical velocity logs shown in Figure 3.15

(b) Initial smooth model $\mathbf{v}_{\mathrm{smth}} = 1/\sqrt{\mathbf{m}_{\mathrm{smth}}}$

(c) Result on the full data with all shots and frequencies after 100 iterations

Figure 3.8: The BG-Compass model with velocity range of 1500 to 4500 m/s, the initial model and the FWI results using the full data



(a) Original model $\mathbf{v}_{\text{true}} = 1/\sqrt{\mathbf{m}_{\text{true}}}$; blue triangles mark horizontal positions for vertical velocity logs shown in Figure 3.16

(b) Initial smooth model $\mathbf{v}_{\text{smth}} = 1/\sqrt{\mathbf{m}_{\text{smth}}}$

(c) Result on the full data with all shots and frequencies after 100 iterations

Figure 3.9: The Marmousi model with velocity range of 1500 to 5800 m/s, the initial model and the FWI results using the full data

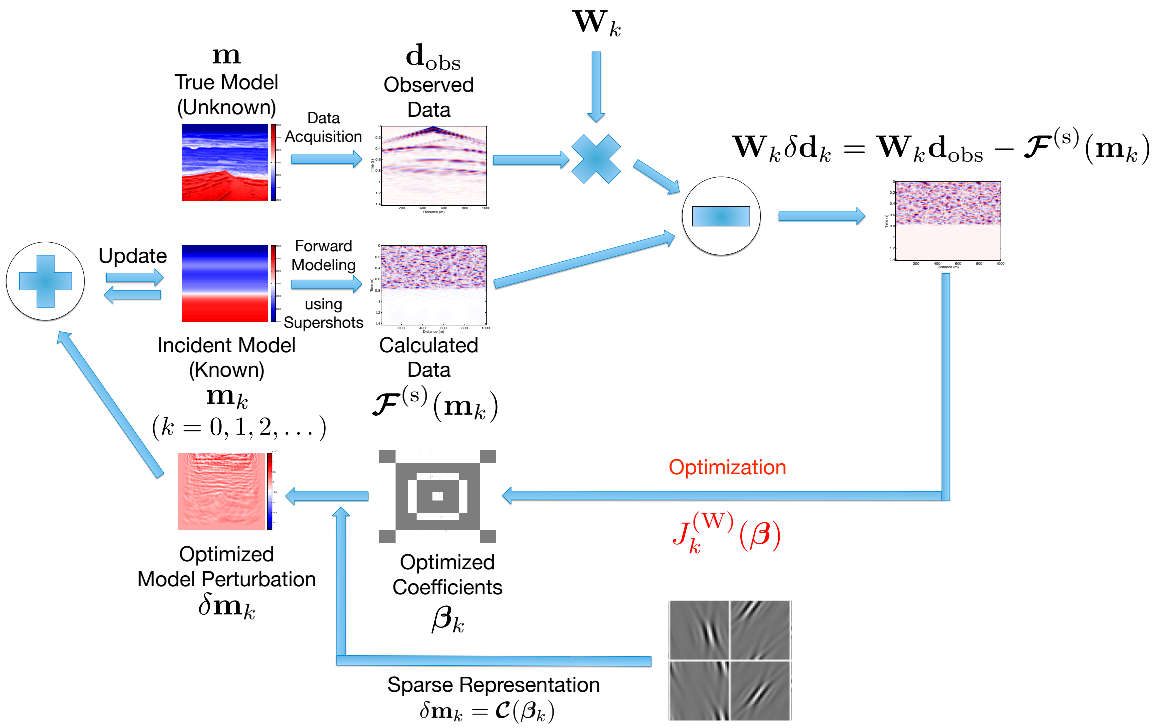
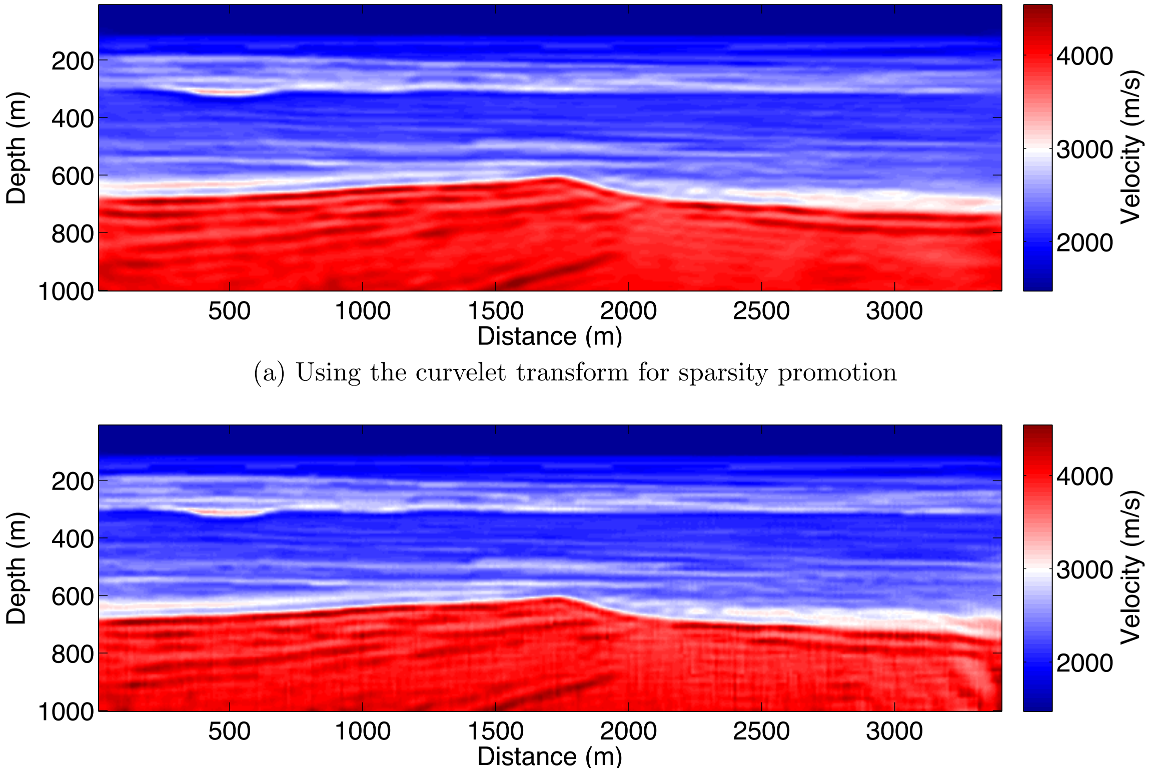

$\mathbf{W}_k$
$\mathbf{m}$
True Model
(Unknown)
Data
Acquisition
$\mathbf{d}_{\mathrm{obs}}$
Observed
Data
$\mathbf{W}_k \delta \mathbf{d}_k = \mathbf{W}_k \mathbf{d}_{\mathrm{obs}} - \boldsymbol{\mathcal{F}}^{(\mathrm{s})}(\mathbf{m}_k)$
Update
Forward
Modeling
using
Supershots
Incident Model
(Known)
$\mathbf{m}_k$
$(k = 0, 1, 2, \dots)$
Calculated
Data
$\boldsymbol{\mathcal{F}}^{(\mathrm{s})}(\mathbf{m}_k)$
Optimization
$J_k^{(\mathrm{W})}(\boldsymbol{\beta})$
Optimized
Model Perturbation
$\delta \mathbf{m}_k$
Optimized
Coefficients
$\boldsymbol{\beta}_k$
Sparse Representation
$\delta \mathbf{m}_k = \boldsymbol{\mathcal{C}}(\boldsymbol{\beta}_k)$

Figure 3.10: FWI workflow using curvelet-domain sparsity promotion with a fixed transform $\mathcal{C}$

are provided in Figures 3.11(a) and 3.12(a), respectively, where the ℓ_1-norm sparsity regularizations on $\boldsymbol{\beta}$ are imposed during the l-PQN iterations.

3.5.2 Sparsity Regularization using SOT

The orthonormal dictionaries $\mathbf{D}_k$ learned for SOT within the five frequency bands are visualized as follows. Since the $\delta\mathbf{m}$ inverted in different frequency bands contain different wavenumber components and their features have different scales, the online orthonormal dictionary learning algorithm is reinitialized from $k = 0$ with a Discrete Cosine Transform (DCT) orthonormal dictionary $\mathbf{D}_0$ every time FWI moves forward to a new frequency band. For the case of the BG-Compass model, the default size of the training patches from $\delta\mathbf{m}$ is $n_z \times n_x = 20 \times 20$, $N = n_z n_x = 400$, so that the dictionaries $\mathbf{D}_k \in \mathbb{R}^{400 \times 400}$. Similarly, for the Marmousi model, the default size of training patches from $\delta\mathbf{m}$ is $n_z \times n_x = 16 \times 24$, $N = n_z n_x = 384$, so that the dictionaries $\mathbf{D}_k \in \mathbb{R}^{384 \times 384}$. Figures 3.13 and 3.14 show how the dictionaries evolve by Algorithm 3.1 during FWI iterations on different frequency bands, in which each $n_z \times n_x$ patch of $\delta\mathbf{m}$ is a linear combination of the atoms visualized as blocks. Figures 3.13(a) and 3.14(a) are the DCT dictionaries $\mathbf{D}_0$ that initialize Algorithm 3.1 when FWI starts processing a new frequency band. After completing $K = 20$ iterations of FWI as well as online dictionary learning, Figures 3.13(b) – 3.13(f) and 3.14(b) – 3.14(f) show the updated orthonormal dictionaries $\mathbf{D}_K$ in each frequency band. The Lagrange multiplier is empirically set as $\lambda = 0.2^2$ for both models so that an appropriate trade-off can be kept between speed of convergence and capability of sparse representation.

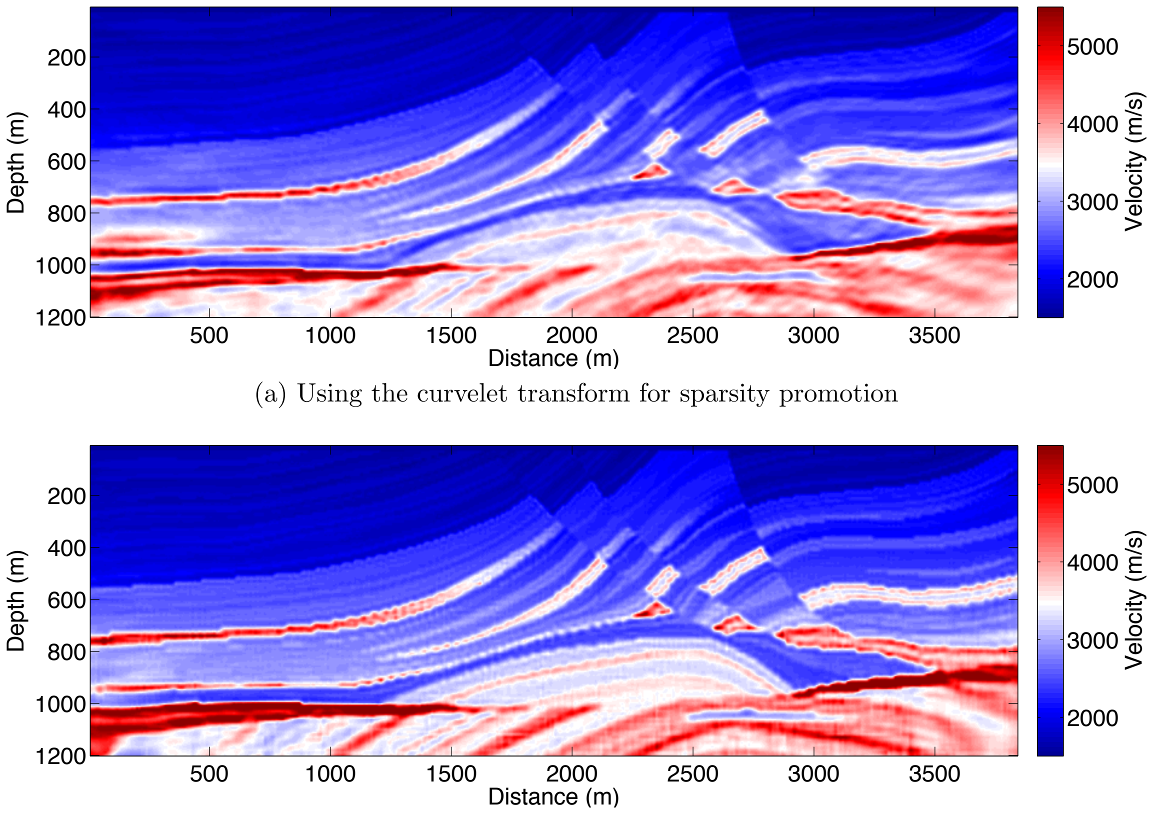


(a) Using the curvelet transform for sparsity promotion

(b) Using SOT for sparsity promotion

Figure 3.11: Compressive FWI results with 3 supershots for the BG-Compass model after 100 iterations

The results of compressive FWI using SOT with the default patch size for sparsity promotion are shown in Figures 3.11(b) and 3.12(b) for BG-Compass and Marmousi, respectively. Different patch sizes $N = n_z \times n_x$ are also tested for dictionaries $\mathbf{D}_k \in$



(a) Using the curvelet transform for sparsity promotion

(b) Using SOT for sparsity promotion

Figure 3.12: Compressive FWI results with 3 supershots for the Marmousi model after 100 iterations

$\mathbb{R}^{N\times N}$ with different sizes so that the robustness of the method can be studied (see Figures 3.15, 3.16, and 3.17).

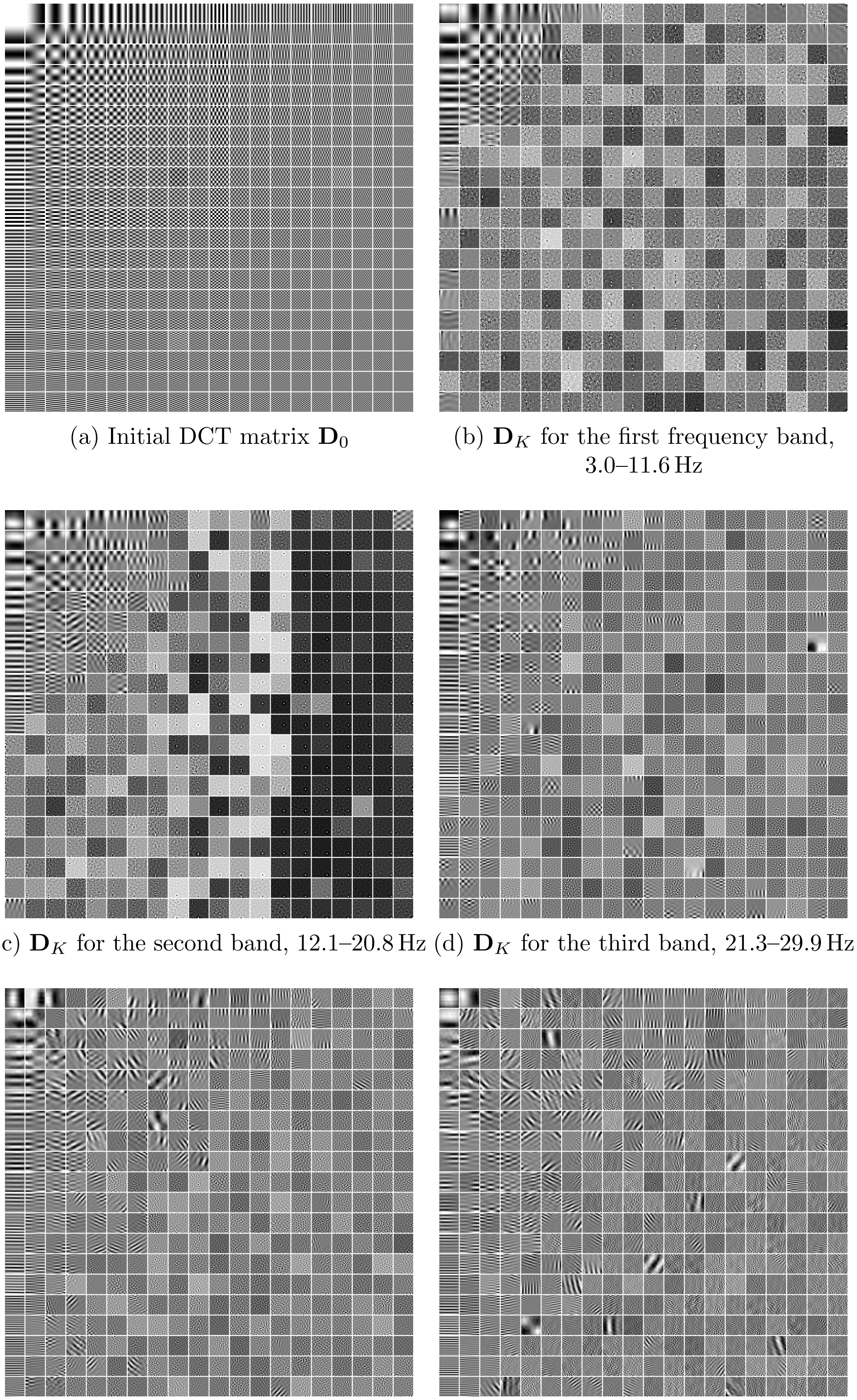
Figures 3.15 and 3.16 show vertical velocity logs for several lateral positions x on the inverted models, marked by blue triangles underneath Figures 3.8(a) and 3.9(a). Besides traditional vertical velocity logs, the quality of FWI can also be measured by the following model fit metric proposed in [52]

$$M(k) \triangleq \left(1 - \frac{\|\mathbf{v}_{\text{true}} - \mathbf{v}_k\|_2}{\|\mathbf{v}_{\text{true}}\|_2}\right) \times 100\% \tag{3.49}$$

where $\mathbf{v}_{\text{true}} = 1/\sqrt{\mathbf{m}_{\text{true}}}$ is the exact velocity model and $\mathbf{v}_k = 1/\sqrt{\mathbf{m}_k}$ is the intermediate velocity model obtained at the k-th FWI iteration. The curves in Figure 3.17 compare the model fit metric $M(k)$ versus FWI iteration number for both velocity models. These results indicate that different patch sizes yield very similar curves and suggest choosing a moderate patch size $N = n_z \times n_x$ that it is neither too huge to train nor too small to represent. It is worth emphasizing that these results give strong evidence that the proposed SOT based on the dictionary learning method can produce inverted models with better visual quality and a higher performance metric than the curvelet transform under the same subsampling ratio for FWI.

To further test the robustness of the method, a noisy seismic dataset is created by adding white Gaussian noise (WGN), and then FWI is performed without any prior denoising process. Figure 3.18 illustrates the wavefield generated by a supershot with the frequency $\omega/(2\pi) = 22.8\,\text{Hz}$, where WGN is added such that the average signal to noise ratio (SNR) equals to $10\,\text{dB}$. As before, $N_s' = 3$ supershots and $N_\omega' = 16$ random frequencies are used for each frequency band.

In this noisy setting of the compressive FWI, both the curvelet transform and the SOT are used for the sparsity promotion of $\delta\mathbf{m}$. Figures 3.19 and 3.20 show the FWI results based on the noisy data for both velocity models, followed by curves showing the model fit metric $M(k)$ versus FWI iterations in Figure 3.21. The results indicate that the SOT can achieve better inverted models than the curvelet transform.



(a) Initial DCT matrix $\mathbf{D}_0$

(b) $\mathbf{D}_K$ for the first frequency band, 3.0–11.6 Hz

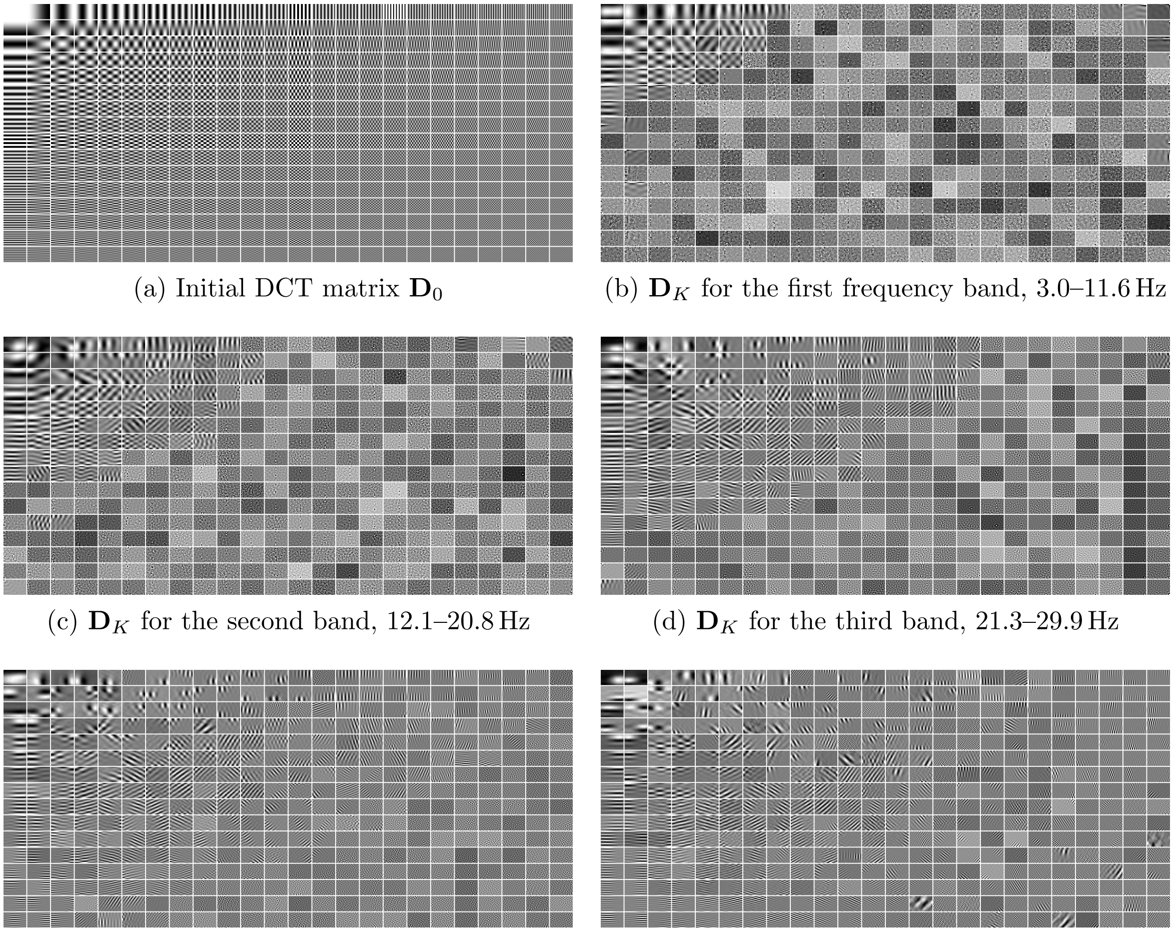
(c) $\mathbf{D}_K$ for the second band, 12.1–20.8 Hz

(d) $\mathbf{D}_K$ for the third band, 21.3–29.9 Hz

(e) $\mathbf{D}_K$ for the fourth band, 30.4–39.0 Hz

(f) $\mathbf{D}_K$ for the fifth band, 39.5–48.1 Hz

Figure 3.13: Initial dictionary $\mathbf{D}_0$ and the learned dictionaries $\mathbf{D}_K$ by Algorithm 3.1 after $K = 20$ FWI iterations in each frequency band on the BG-Compass model. Dictionary size is 400×400; each atom is visualized as a 20×20 block in the images.



(a) Initial DCT matrix $\mathbf{D}_0$

(b) $\mathbf{D}_K$ for the first frequency band, 3.0–11.6 Hz

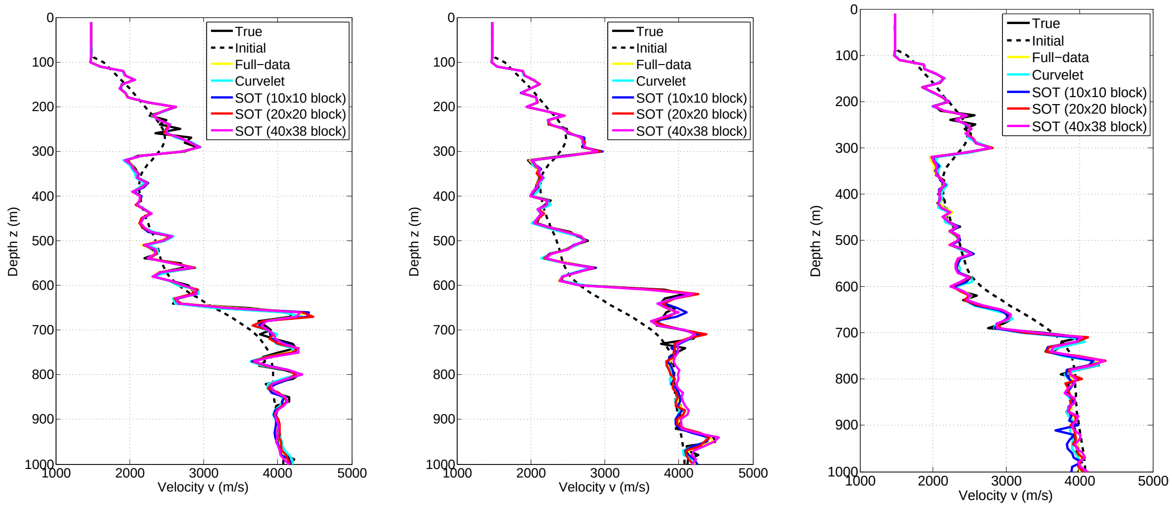
(c) $\mathbf{D}_K$ for the second band, 12.1–20.8 Hz

(d) $\mathbf{D}_K$ for the third band, 21.3–29.9 Hz

(e) $\mathbf{D}_K$ for the fourth band, 30.4–39.0 Hz

(f) $\mathbf{D}_K$ for the fifth band, 39.5–48.1 Hz

Figure 3.14: Initial dictionary $\mathbf{D}_0$ and the learned dictionaries $\mathbf{D}_K$ by Algorithm 3.1 after $K = 20$ FWI iterations in each frequency band on the Marmousi model. Dictionary size is 384×384; each atom is visualized as a 16×24 block in the images.

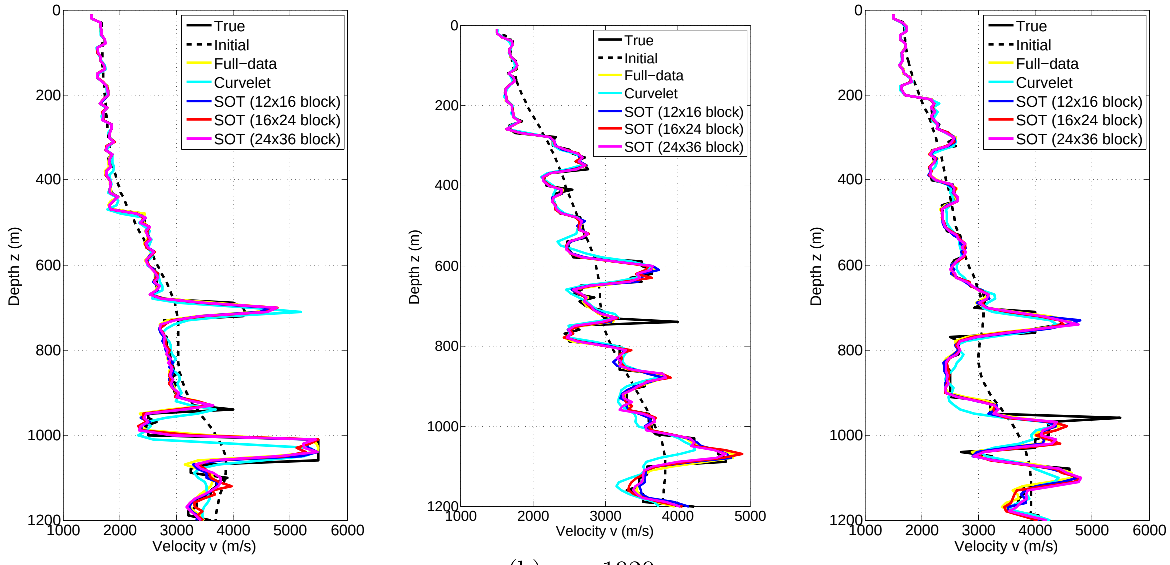


(a) $x = 875\,\mathrm{m}$

(b) $x = 1750\,\mathrm{m}$

(c) $x = 2625\,\mathrm{m}$

Figure 3.15: Vertical velocity logs for the BG-Compass model. Three different patch sizes are tested.

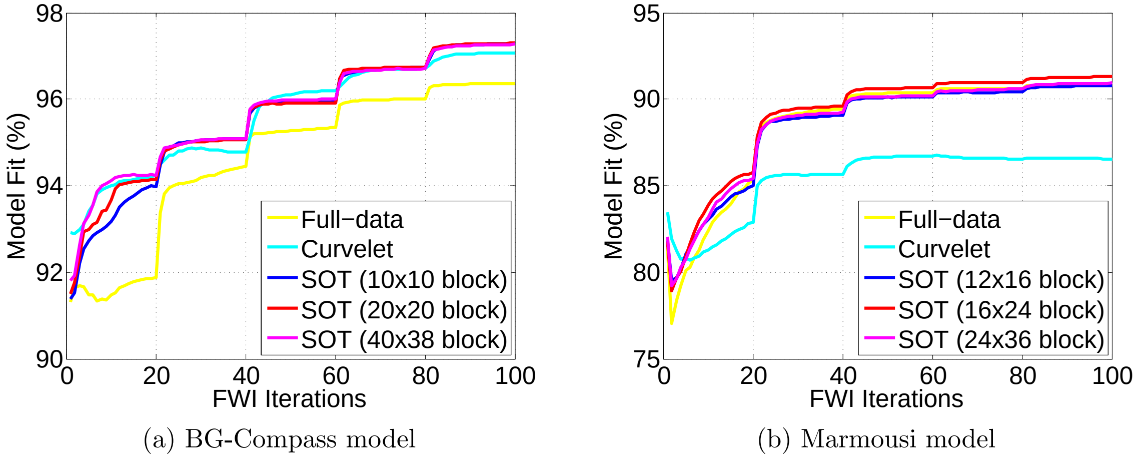


(a) $x = 960\,\mathrm{m}$

(b) $x = 1920\,\mathrm{m}$

(c) $x = 2880\,\mathrm{m}$

Figure 3.16: Vertical velocity logs for the Marmousi model. Three different patch sizes are tested.



(a) BG-Compass model

(b) Marmousi model

Figure 3.17: Model fit versus FWI iteration number for SOT-domain sparsity regularization. Three different patch sizes are tested.

Meanwhile, comparing Figures 3.19(b) and 3.20(b) with Figures 3.11(b) and 3.12(b), respectively, good FWI results can still be obtained with noisy data, which results from the SOT-domain sparsity regularization.

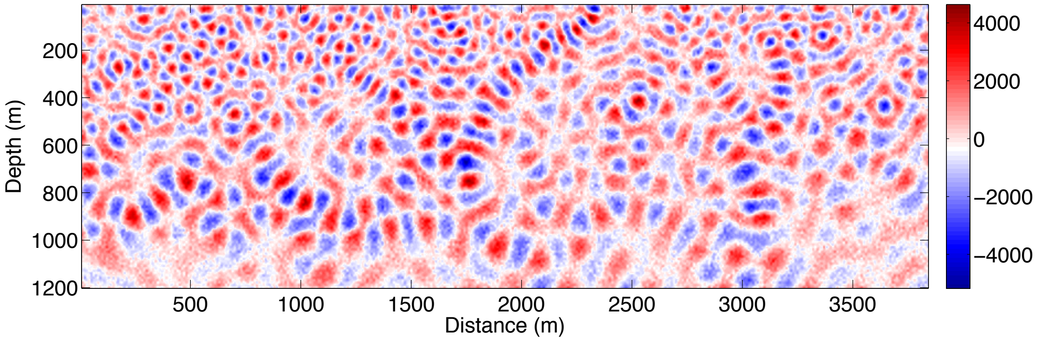
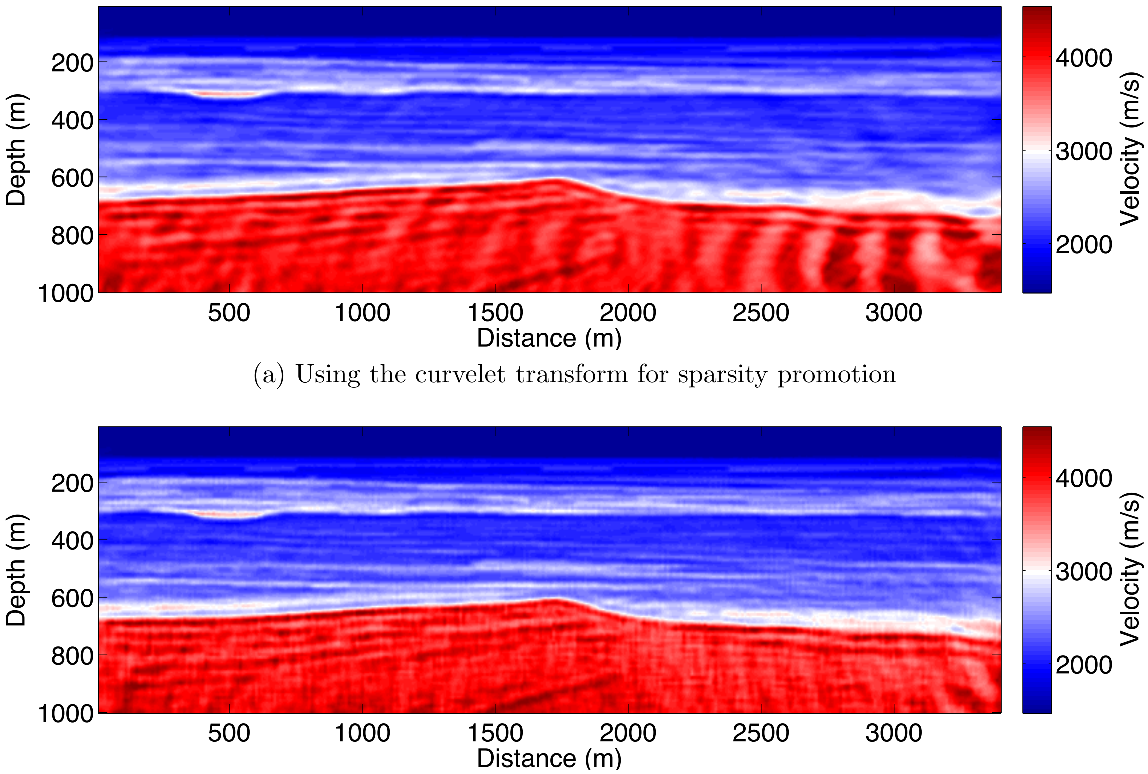


Figure 3.18: Noisy wavefield examples generated by a supershot with frequency 22.8 Hz, SNR = 10 dB

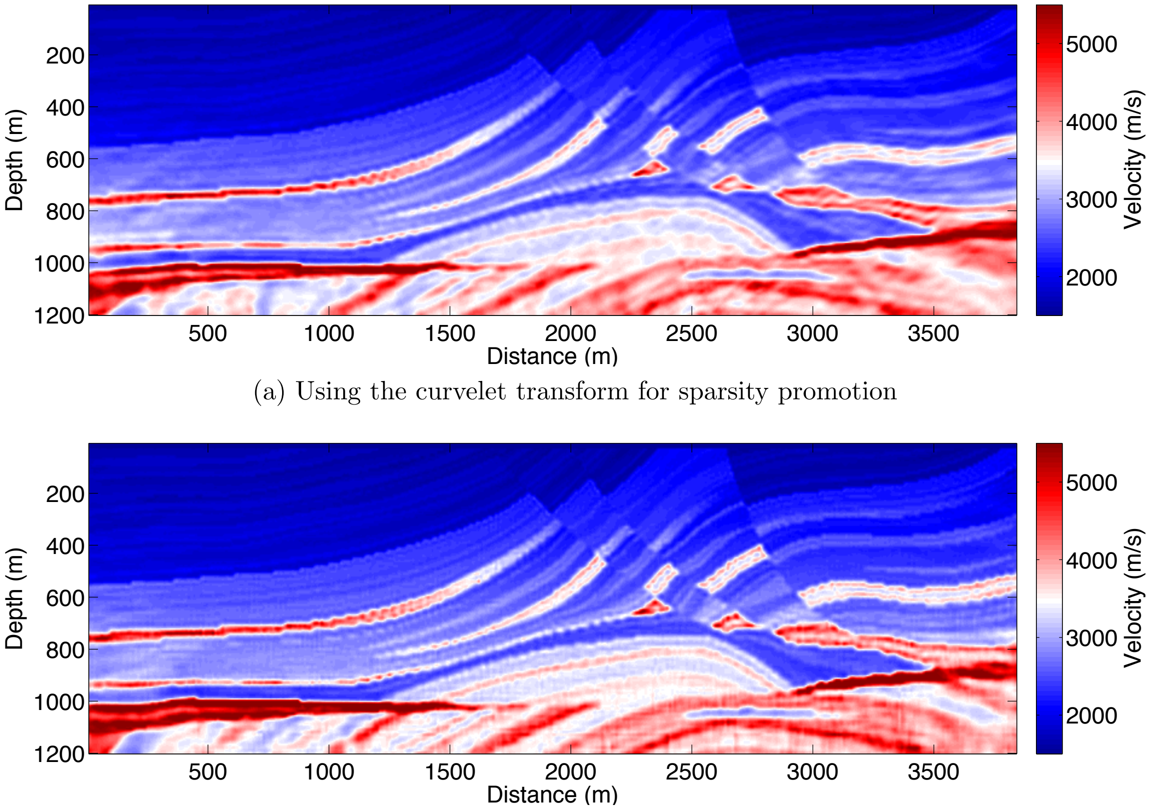
The extra computational overhead involved in learning the orthonormal dictionary for SOT has been analyzed in Section 3.3. In addition to the theoretical complexity analysis, it is useful to report one instance of the actual running time of forward modeling, l-PQN optimization and orthonormal dictionary learning (with 16×24 patches) for 20 compressive FWI iterations (in one frequency band) on the Marmousi model with 3 supershots and 16 random frequencies. As a comparison, the running



(a) Using the curvelet transform for sparsity promotion

(b) Using SOT for sparsity promotion

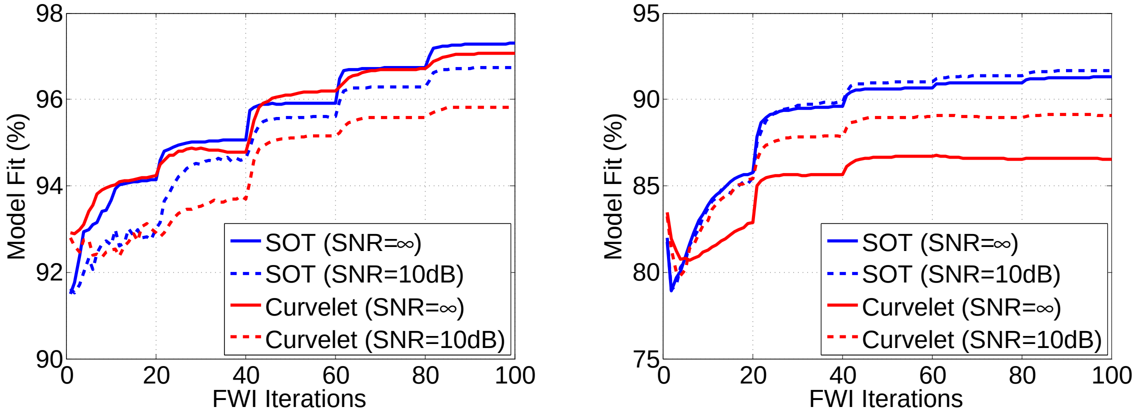
Figure 3.19: Compressive FWI results with 3 supershots for the BG-Compass model after 100 iterations; input data is noisy with average SNR $= 10\,\mathrm{dB}$.



(a) Using the curvelet transform for sparsity promotion

(b) Using SOT for sparsity promotion

Figure 3.20: Compressive FWI results with 3 supershots for the Marmousi model after 100 iterations; input data is noisy with average SNR = 10 dB.



(a) BG-Compass model

(b) Marmousi model

Figure 3.21: Model fit versus FWI iteration number of SOT-domain sparsity regularization with noiseless data (blue line) and noisy data at average SNR = 10 dB (red line)

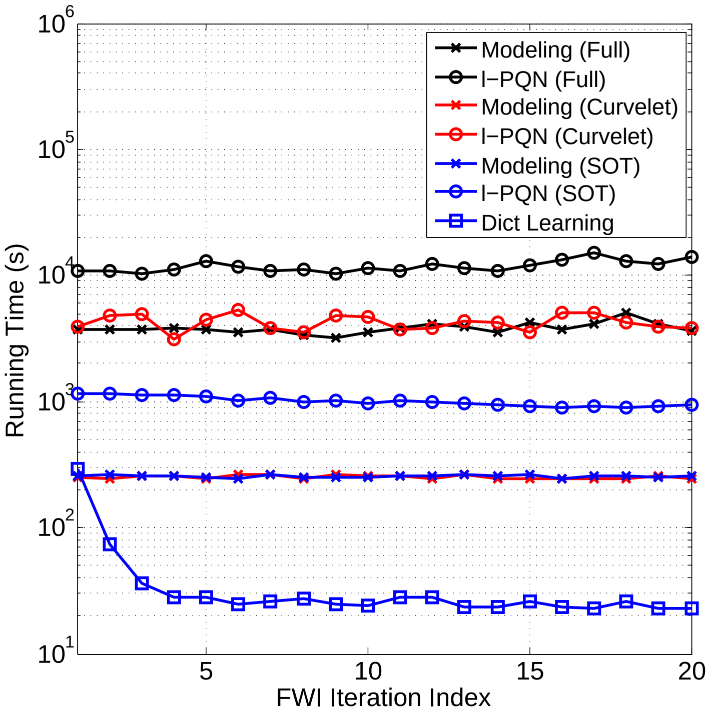


Figure 3.22: Running time profile of forward modeling, l-PQN optimization and dictionary learning versus FWI iterations

times of forward modeling and l-PQN optimization for the full-data FWI are also provided. These profiles are shown in Figure 3.22. The computing cluster used for time profiling is based on 12-core Intel® Xeon® CPU with 64GB RAM. Both the forward modeling and the l-PQN optimization are accelerated by parallel computing.

Figure 3.22 shows that the running time of forward modeling and l-PQN optimization for the full-data FWI is over 10 times of that for the compressive FWI. It is also noticeable that, after the first FWI iteration, the running time for orthonormal dictionary learning falls rapidly to a negligible level compared to the cost of the other two phases. The online learning approach exhibits this behavior because it always updates the latest and best dictionary for the incoming model perturbation at each FWI iteration. Once a good dictionary has been obtained, many fewer training iterations are required for the updates. Therefore, it is safe to say that the actual overhead from orthonormal dictionary learning is not significant.

Figure 3.22 also shows the running time of forward modeling and l-PQN optimization of compressive FWI using the curvelet transform for sparsity promotion. The modeling time is almost the same as the SOT-based method since the compression ratio is exactly the same. The l-PQN optimization time of the curvelet-based method is several times slower than the SOT-based method. The underlying reason is not difficult to explain, as the curvelet transforms, especially for those with high decomposition levels on scales and orientations, have a much larger redundancy ratio (about 7.2 when curvelets are used at the finest scale [20]) than the SOT (exactly 1 if evenly decomposed into patches, or slightly larger than 1 if unevenly decomposed).

For example, the Marmousi model of size $N_z \times N_x = 120 \times 384$ has $144 \times 432 = 62208$ grids after adding the perfectly matched layer of thickness 24 on left, right and bottom. Its curvelet transform coefficient vector with 5 decomposition scales and 32, 64, 64, 128 angles on the 2nd to 5th scales, respectively, is of length 476450. Hence the redundancy ratio is $476450/62208 = 7.66$. On the other hand, the SOT coefficient vector with an uneven patch decomposition is of length 65664, and hence the redundancy ratio is $65664/62208 = 1.06$. Since many vector and matrix calculations are involved in the l-PQN optimization in which some vectors and matrices are as tall as the coefficient vector (see Algorithm 3.2 and Equations (3.43), (3.48), etc.), the difference in computational time between the curvelet-based and SOT-based methods can be found easily. As a final comment, it is possible to reduce the redundancy ratio of the curvelet transform as well as the l-PQN optimization time by using simpler transforms with less amount of curvelets, however the performance of the compressive FWI with curvelets would be degraded.

CHAPTER IV

CONCLUSIONS AND FUTURE EXTENSIONS

4.1 Conclusions

Based on the study of wave propagation characteristics under the surface of the earth, seismic methods are extensively used in geophysical exploration. These methods acquire seismic data from an array of receivers deployed in the seismic survey area, and obtain a subsurface image of the earth by means of seismic imaging. Therefore, high-quality seismic data reconstruction has become a critical preprocessing step prior to the standard seismic imaging techniques such as migration and inversion. With well-reconstructed seismic data, FWI is able to estimate high-resolution subsurface velocity models. However, FWI needs to use a great amount of seismic data and has a very high computational complexity. One way to improve its efficiency is by incorporating compressive sensing techniques to reduce its internal data dimensionality by exploiting sparse representation and approximation of signals.

Dictionary learning has now become a promising technique for sparse signal representation and approximation. Compared to traditional transforms such as wavelet, contourlet, curvelet, etc. with predefined dictionaries, dictionary learning methods are better able to adapt to nonintuitive signal regularities beyond piecewise smoothness and can generate sparser signal representations. The key idea of dictionary learning is that the dictionary has to be inferred from a set of training signals, which can be either an outside corpus or the signals generated during processing. In this thesis, the latter category is chosen for dictionary learning as it would be impractical to obtain a large set of seismic signals from outside sources due to many restrictions.

In Chapter 2 of the thesis, I presented novel reconstruction techniques, including denoising and inpainting, for seismic datasets based on a sparsity-promoting dictionary learning method. Unlike previous methods that train fully explicit dictionary matrices but sacrifice efficiency, this method only requires learning a sparse matrix after choosing a base dictionary that corresponds to an efficient transform and also incorporates some prior knowledge about the data. Moreover, motivated by the underlying structural similarity among dictionary atoms, this method involves a constraint that each atom in the learned dictionary is itself a linear combination of atoms in the base dictionary. Such a method improves the efficiency and stability of dictionary learning and provides a new layer of adaptivity to the existing efficient transforms. The experimental results indicate that both denoising and inpainting results significantly outperform traditional methods based on the fixed transforms.

In Chapter 3 of the thesis, I presented a novel and efficient compressive sensing scheme that significantly reduces the computational complexity for FWI. The new method exploits sparsity by representing model perturbations with a sparse orthonormal transform (SOT) such that each patch of the model perturbation can be represented with sparse coefficients over adaptive data-driven dictionaries trained from previous results. Compared to traditional fixed transforms that are only optimal for objects with piecewise smoothness, the SOT is better able to adapt to nonintuitive signal regularities such as complex geophysical features. Compared to the traditional overcomplete dictionary learning methods, the orthonormal dictionary learning method is much more efficient and can work in an online manner. The SOT enables a significant reduction in the amount of data used in FWI by invoking the strategy of compressive sampling, which is implemented by generating a few supershots and selecting a small number of frequencies for forward modeling. After that, the original Gauss-Newton problem becomes an LASSO problem which can be effectively solved using a projected quasi-Newton algorithm. The experiments presented show that high-quality inverted velocity models can be obtained with both simple and complex geophysical features by working with a small subset of the full seismic dataset, even in the presence of noise.

4.2 Future Extensions

Future work based on this research, especially for the efficient FWI implementation using the SOT-based FWI, could be undertaken into two ways. The first one is to extend the 2D frequency-domain FWI into 3D. Solving 3D FWI problems is much more expensive than its 2D counterpart. In 2D FWI problems, the velocity models are of size $N_z \times N_x$, which may include tens of thousands of grid points. However, in 3D FWI problems there is another lateral direction, the y-axis, such that velocity models are of size $N_z \times N_x \times N_y$ with millions of grid points. The seismic wave equation has to be propagated over many more grid points in 3D models, so the computational cost will grow very rapidly with the model size. This creates a situation where one can learn 3D-atom dictionaries and use them to produce a sparse representation of the 3D model perturbations. Since dictionary learning reshapes model patches and atoms into vectors no matter how many dimensions in space they occupy, one can reasonably infer their computational complexity according to the learning steps in Algorithm 3.1. If the 3D patch size is $n_z \times n_x \times n_y$, where $n_z \ll N_z$, $n_x \ll N_x$, $n_y \ll N_y$, and all possible overlapping patches are used for training, then each orthonormal dictionary learning iteration costs $\mathcal{O}((n_z n_x n_y)^2 (N_z - n_z + 1)(N_x - n_x + 1)(N_y - n_y + 1) + (n_z n_x n_y)^3)$, and applying the SOT with the 3D dictionary to a 3D model costs $\mathcal{O}(n_z n_x n_y N_z N_x N_y)$. Because the 3D geometry permits extra freedom for subsampling, the compressive sensing scheme presented in the thesis could reduce even more the dimensionality in 3D FWI problems.

The second avenue for future research is to extend the frequency-domain FWI using random source encoding and SOT-based sparsity promotion into the time domain.

Although FWI problems in both the time and frequency domains minimize almost the same least-squares misfit function between the recorded and modeled seismic data, implementations can be quite different. In the frequency domain, the gradient vector and Hessian matrix of the FWI misfit function are computed with the help of monochromatic Green's functions. In the time domain, the gradient of the FWI misfit function can be constructed by cross-correlating the forward modeling wavefield from a shot source with a backward modeling wavefield from data residuals [65, 127]. Time-domain FWI takes all frequencies into account for inversion, and it can yield more accurate results. In addition, it costs less memory than the frequency-domain FWI since no Helmholtz operator matrix needs to be inverted. However, time-domain FWI could take significant computation time when the time step is small or the simulation time duration is long. Future research work could generate supershots directly in the time domain to reduce the problem dimensionality and perform online orthonormal dictionary learning on the patches of the optimized model perturbations in previous time-domain FWI iterations to build adaptive SOTs for sparsity promotion.

APPENDIX A

THE PARALLEL MATRIX-FREE FRAMEWORK FOR SEISMIC SIMULATION, SURVEY AND IMAGING

A.1 Introduction

Numerical simulation of seismic wave propagation is the cornerstone of geophysical exploration. Large amounts of seismic data must be acquired in order to estimate subsurface properties for the purposes of academic research and industrial production. Every successful seismic inversion software framework consists of a seismic wave modeling engine that solves wave equations with model parameters to generate seismic data, and an optimization engine that updates the model parameters based on the value, gradient and Hessian matrix of the data-misfit objective function. With the rapidly increasing need for exploring geologically more complex subsurface areas, computation of seismic wave modeling and model parameter optimization have come to heavily rely on high performance computing (HPC).

In order to facilitate the development of new seismic inversion methods, solving seismic equations has to be well encapsulated as robust, efficient and scalable software modules. This appendix chapter introduces basic concepts of wave propagation using finite difference method in both time and frequency domains.

A.2 Acoustic and Elastic Wave Equations

The earth is an elastic media such that the seismic body waves traveling through the interior of the earth have two components: primary wave (P-wave) and secondary wave (S-wave). P-waves arrive at receivers first as they travel faster than any other waves. P-waves are also called pressure waves as they cause pressure vibrations formed by alternating from compression to expansion of the medium along the wave traveling direction. Hence P-waves are a type of longitudinal wave. S-waves travel slower than P-waves and are also called shear waves as they shear the medium instead of changing the volume of the medium through which they propagate. Hence S-waves are a type of transverse wave.

For simplicity, some industrial seismic processing only considers P-waves, which are described by an acoustic wave equation. It is verified by borehole data that the density variations of the medium are not the main source of reflected waves [59]. Therefore, it is usually safe to assume a constant density of the medium. Then the acoustic wave equation can be written as

$$\frac{1}{v^2(\mathbf{x})}\frac{\partial^2 p(\mathbf{x},t)}{\partial t^2} - \nabla^2 p(\mathbf{x},t) = f(\mathbf{x},t) \tag{A.1}$$

where $\mathbf{x} \triangleq (x, y, z)$ is the 3D Cartesian coordinates in which x, y are two lateral coordinates and z is the vertical coordinate, $v(\mathbf{x})$ is the velocity of acoustic wave, and $p(\mathbf{x}, t)$ is the acoustic pressure wavefield. The Laplace operator is defined as $\nabla^2 \triangleq \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$. On the right-hand side of (A.1), $f(\mathbf{x}, t)$ is the source function that provides the initial wave energy. The Ricker wavelet is widely used as a seismic source function and its time-domain function can be written as

$$f(t) = (1 - 2\pi^2 f_{\mathrm{p}}^2 t^2)e^{-\pi^2 f_{\mathrm{p}}^2 t^2} \tag{A.2}$$

where f_{p} refers to the peak frequency.

Elastic wave modeling offers a more realistic simulation approach than acoustic wave modeling to study seismic wave propagation in the earth. Using a compact form, the full three-component elastic wave equation can be written as

$$\frac{\partial^2 \mathbf{s}}{\partial t^2} = v_{\mathrm{p}}^2 \nabla (\nabla \cdot \mathbf{s}) - v_{\mathrm{s}}^2 \nabla \times (\nabla \times \mathbf{s}) + \mathbf{f} \tag{A.3}$$

where $\mathbf{s} \triangleq [s_x, s_y, s_z]^T$ is the vector wavefield of particle displacement in the 3D Cartesian coordinate system (x, y, z), v_{p} is the P-wave velocity, and v_{s} is the S-wave velocity. The vector $\mathbf{f} \triangleq [f_x, f_y, f_z]^T$ is the 3D input source. Since P-waves and S-waves are coupled in $\mathbf{s}$, a separation of two wave types is necessary if one needs to process each type independently with tailored algorithms. The Helmholtz decomposition [3] can decouple the vector wavefield $\mathbf{s}$ in (A.3) (ignoring the 3D input source $\mathbf{f}$) into a curl-free wavefield $\mathbf{s}_\mathrm{p}$ and a divergence-free wavefield $\mathbf{s}_\mathrm{s}$ for P-waves and S-waves, respectively, that satisfy

$$\begin{cases} \dfrac{\partial^2 \mathbf{s}_\mathrm{p}}{\partial t^2} = v_\mathrm{p}^2 \nabla\left(\nabla \cdot \mathbf{s}\right) = v_\mathrm{p}^2 \nabla A \\ \dfrac{\partial^2 \mathbf{s}_\mathrm{s}}{\partial t^2} = -v_\mathrm{s}^2 \nabla \times \left(\nabla \times \mathbf{s}\right) = -v_\mathrm{s}^2 \nabla \times \mathbf{B} \end{cases} \tag{A.4}$$

where $\mathbf{s}_\mathrm{p} \triangleq [s_{x\mathrm{p}}, s_{y\mathrm{p}}, s_{z\mathrm{p}}]^T$, $\mathbf{s}_\mathrm{s} \triangleq [s_{x\mathrm{s}}, s_{y\mathrm{s}}, s_{z\mathrm{s}}]^T$ are the particle displacement vectors caused by P-waves and S-waves, respectively. $A \triangleq \nabla \cdot \mathbf{s}$ and $\mathbf{B} \triangleq \nabla \times \mathbf{s} = [B_1, B_2, B_3]^T$ are auxiliary variables. Since particle velocity $\mathbf{v} \triangleq \dfrac{\partial \mathbf{s}}{\partial t} = \mathbf{v}_\mathrm{p} + \mathbf{v}_\mathrm{s}$ where $\mathbf{v} \triangleq [v_x, v_y, v_z]^T$, $\mathbf{v}_\mathrm{p} \triangleq [v_{x\mathrm{p}}, v_{y\mathrm{p}}, v_{z\mathrm{p}}]^T$ and $\mathbf{v}_\mathrm{s} \triangleq [v_{x\mathrm{s}}, v_{y\mathrm{s}}, v_{z\mathrm{s}}]^T$, for each component, (A.4) can be expanded as a series of equivalent first-order elastic wave equations

$$\begin{cases} \dfrac{\partial v_{x\mathrm{p}}}{\partial t} = v_\mathrm{p}^2 \dfrac{\partial A}{\partial x}, \quad \dfrac{\partial v_{y\mathrm{p}}}{\partial t} = v_\mathrm{p}^2 \dfrac{\partial A}{\partial y}, \quad \dfrac{\partial v_{z\mathrm{p}}}{\partial t} = v_\mathrm{p}^2 \dfrac{\partial A}{\partial z} \\ \dfrac{\partial v_{x\mathrm{s}}}{\partial t} = -v_\mathrm{s}^2 \left(\dfrac{\partial B_3}{\partial y} - \dfrac{\partial B_2}{\partial z}\right) \\ \dfrac{\partial v_{y\mathrm{s}}}{\partial t} = -v_\mathrm{s}^2 \left(\dfrac{\partial B_1}{\partial z} - \dfrac{\partial B_3}{\partial x}\right) \\ \dfrac{\partial v_{z\mathrm{s}}}{\partial t} = -v_\mathrm{s}^2 \left(\dfrac{\partial B_2}{\partial x} - \dfrac{\partial B_1}{\partial y}\right) \end{cases} \tag{A.5}$$

where the auxiliary variables A, B_1, B_2 and B_3 are updated by

$$\begin{cases} \dfrac{\partial A}{\partial t} = \dfrac{\partial v_x}{\partial x} + \dfrac{\partial v_y}{\partial y} + \dfrac{\partial v_z}{\partial z} \\ \dfrac{\partial B_1}{\partial t} = \dfrac{\partial v_z}{\partial y} - \dfrac{\partial v_y}{\partial z} \\ \dfrac{\partial B_2}{\partial t} = \dfrac{\partial v_x}{\partial z} - \dfrac{\partial v_z}{\partial x} \\ \dfrac{\partial B_3}{\partial t} = \dfrac{\partial v_y}{\partial x} - \dfrac{\partial v_x}{\partial y}. \end{cases} \tag{A.6}$$

A.3 The Finite-difference Time-domain (FDTD) Method

Finite difference (FD) methods are widely used to solve wave equations numerically because they are easy to implement, and have high accuracy as well as efficiency

[4, 31]. In order to solve the wave equation with a FD method, the continuous functions and velocity models are represented by their values at grid points and derivatives are approximated by linear combination of these values. Instead of solving the wave equation in a continuous domain analytically, FD methods provide an approximated solution on these grid points.

The estimation of derivatives used in wave equations (A.1), (A.5) and (A.6) are crucial for FD methods. By writing $p(\mathbf{x}, t)$ in full as $p(x, y, z, t)$, the partial derivative of $p(x, y, z, t)$ with respect to, for example, x, is defined as

$$\frac{\partial p(x,y,z,t)}{\partial x} = \lim_{\Delta x \to 0} \frac{p(x+\Delta x, y, z, t) - p(x,y,z,t)}{\Delta x}, \tag{A.7}$$

and can be approximated by a scaled difference

$$\frac{\partial p(x,y,z,t)}{\partial x} \approx \frac{p(x+\Delta x, y, z, t) - p(x,y,z,t)}{\Delta x} \tag{A.8}$$

assuming that the grid spacing Δx is a small finite value rather than infinitesimal. After applying the approximation twice, a central finite difference scheme for the second-order partial derivative can be approximated as

$$\begin{aligned} \frac{\partial^2 p(x,y,z,t)}{\partial x^2} &= \frac{\partial}{\partial x}\left(\frac{\partial p(x,y,z,t)}{\partial x}\right) \\ &\approx \frac{1}{\Delta x}\left(\frac{\partial p(x,y,z,t)}{\partial x} - \frac{\partial p(x-\Delta x,y,z,t)}{\partial x}\right) \\ &\approx \frac{p(x+\Delta x,y,z,t) - 2p(x,y,z,t) + p(x-\Delta x,y,z,t)}{\Delta x^2}. \end{aligned} \tag{A.9}$$

The estimations of derivatives with respect to other arguments y, z and t follow a similar pattern. For simplicity, a more compact form of notation is introduced

$$\begin{cases} v_{i,j,k} \triangleq v(i\Delta x, j\Delta y, k\Delta z) \\ p_{i,j,k}^{(n)} \triangleq p(i\Delta x, j\Delta y, k\Delta z, n\Delta t) \\ f_{i,j,k}^{(n)} \triangleq f(i\Delta x, j\Delta y, k\Delta z, n\Delta t), \end{cases} \tag{A.10}$$

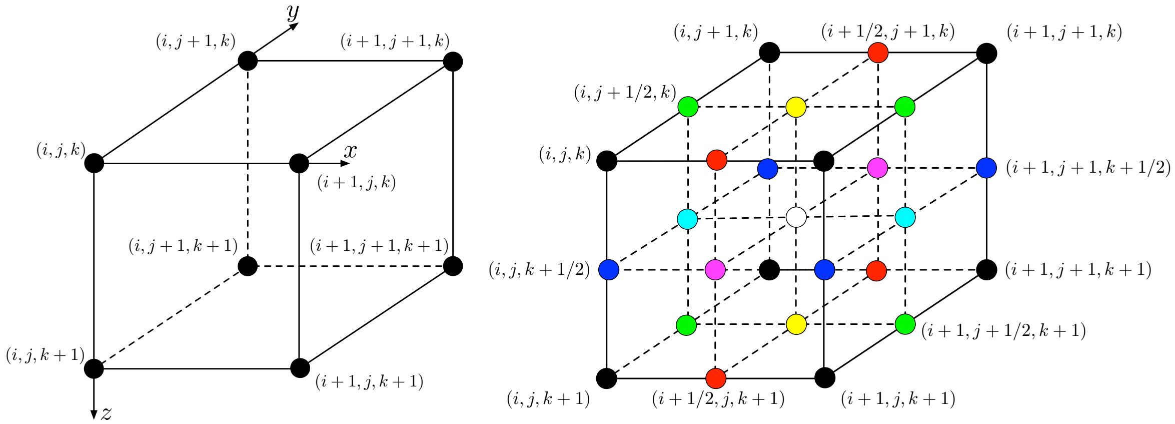
then the FDTD expression of the time-domain acoustic wave equation (A.1) can be written as

$$\frac{1}{v_{i,j,k}^2}\frac{p_{i,j,k}^{(n+1)} - 2p_{i,j,k}^{(n)} + p_{i,j,k}^{(n-1)}}{\Delta t^2} - f_{i,j,k}^{(n)}$$
$$= \frac{p_{i+1,j,k}^{(n)} - 2p_{i,j,k}^{(n)} + p_{i-1,j,k}^{(n)}}{\Delta x^2} + \frac{p_{i,j+1,k}^{(n)} - 2p_{i,j,k}^{(n)} + p_{i,j-1,k}^{(n)}}{\Delta y^2} + \frac{p_{i,j,k+1}^{(n)} - 2p_{i,j,k}^{(n)} + p_{i,j,k-1}^{(n)}}{\Delta z^2}. \tag{A.11}$$

Simple algebraic manipulations lead to an iterative forward update expression for the discretized acoustic pressure wavefield as

$$\begin{aligned} p_{i,j,k}^{(n+1)} &= \frac{v_{i,j,k}^2 \Delta t^2}{\Delta x^2}\left(p_{i+1,j,k}^{(n)} - 2p_{i,j,k}^{(n)} + p_{i-1,j,k}^{(n)}\right) \\ &+ \frac{v_{i,j,k}^2 \Delta t^2}{\Delta y^2}\left(p_{i,j+1,k}^{(n)} - 2p_{i,j,k}^{(n)} + p_{i,j-1,k}^{(n)}\right) \\ &+ \frac{v_{i,j,k}^2 \Delta t^2}{\Delta z^2}\left(p_{i,j,k+1}^{(n)} - 2p_{i,j,k}^{(n)} + p_{i,j,k-1}^{(n)}\right) \\ &+ 2p_{i,j,k}^{(n)} - p_{i,j,k}^{(n-1)} + v_{i,j,k}^2 \Delta t^2 f_{i,j,k}^{(n)}. \end{aligned} \tag{A.12}$$

In (A.12), all values of $p(\mathbf{x}, t)$ are computed on standard integer-grid points which are illustrated as black circles marked in Figure A.1(a). This is an easy-to-understand scheme which serves as an excellent introductory example. However, in order to obtain better accuracy, a staggered-grid scheme needs to be used.



(a) Standard Grid

(b) Staggered Grid

Figure A.1: Grid discretization modes

With a sophisticated design, it turns out that higher-order approximations of the derivatives can be obtained with much reduced approximation error, if one can make use of the half-grid points, which are staggered with respect to the integer-grid points. As a 3D example, Figure A.1(b) illustrates 8 different sets of staggered grids in which 7 of them marked by non-black colors are located in half-grid points.

The Taylor series of a function $p(u)$ on the half-grid points can be written as

$$\begin{cases} p\left(u+\dfrac{2k+1}{2}\Delta u\right) = p(u)+\displaystyle\sum_{n=1}^{\infty}\frac{1}{n!}\frac{\partial^n p(u)}{\partial u^n}\left(\frac{2k+1}{2}\Delta u\right)^n \\ p\left(u-\dfrac{2k+1}{2}\Delta u\right) = p(u)+\displaystyle\sum_{n=1}^{\infty}\frac{(-1)^n}{n!}\frac{\partial^n p(u)}{\partial u^n}\left(\frac{2k+1}{2}\Delta u\right)^n \end{cases} \tag{A.13}$$

where u refers to any one of the x, y, and z axis (denoted as $u = x, y, z$) and $k = 0, 1, 2, \ldots$. The difference between the two lines in (A.13) cancels all the terms with even n, and the result is

$$\begin{aligned} &\frac{p\left(u+\frac{2k+1}{2}\Delta u\right)-p\left(u-\frac{2k+1}{2}\Delta u\right)}{(2k+1)\Delta u} \\ &= \frac{\partial p(u)}{\partial u}+\sum_{n=1}^{\infty}\frac{1}{(2n+1)!}\frac{\partial^{(2n+1)}p(u)}{\partial u^{(2n+1)}}\left(\frac{2k+1}{2}\Delta u\right)^{2n}, \quad k=0,1,2,\ldots \end{aligned} \tag{A.14}$$

Using a linear combination of the finite differences based on (A.14), the partial derivative $\dfrac{\partial p(u)}{\partial u}$ defined at integer grid points $u = k\Delta u$ can be approximated as

$$\begin{aligned} \left.\frac{\partial p(u)}{\partial u}\right|_{u=k\Delta u} &= \sum_{k=0}^{N-1} a_k \frac{p\left(u+\frac{2k+1}{2}\Delta u\right)-p\left(u-\frac{2k+1}{2}\Delta u\right)}{(2k+1)\Delta u} \\ &= \sum_{k=0}^{N-1} a_k\left[\frac{\partial p(u)}{\partial u}+\frac{\Delta u^2}{3!\cdot 2^2}(2k+1)^2\frac{\partial^3 p(u)}{\partial u^3}+\frac{\Delta u^4}{5!\cdot 2^4}(2k+1)^4\frac{\partial^5 p(u)}{\partial u^5}+\cdots\right. \\ &\left.+\frac{\Delta u^{2N-2}}{(2N-1)!\cdot 2^{2N-2}}(2k+1)^{2N-2}\frac{\partial^{2N-1}p(u)}{\partial u^{2N-1}}+o(\Delta u^{2N})\right]. \end{aligned} \tag{A.15}$$

If the weights $\{a_k\}_{k=0}^{N-1}$ are assigned properly, all the terms on the right-hand side of (A.15) can be eliminated except $\dfrac{\partial p(u)}{\partial u}$ and the approximation error in the order of

$o(\Delta u^{2N})$. Therefore, $\{a_k\}_{k=0}^{N-1}$ should satisfy the following system of linear equations

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1^2 & 3^2 & \cdots & (2N-1)^2 \\ 1^4 & 3^4 & \cdots & (2N-1)^4 \\ \vdots & \vdots & \ddots & \vdots \\ 1^{2N-2} & 3^{2N-2} & \cdots & (2N-1)^{2N-2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{N-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

whose results for different values of N can be briefly exemplified as

$$\begin{aligned} N = 1: &\quad a_0 = 1 \\ N = 2: &\quad a_0 = 9/8, a_1 = -1/24 \\ N = 3: &\quad a_0 = 75/64, a_1 = -25/384, a_2 = 3/640 \\ \vdots & \end{aligned}$$

This definition shows that the values of $p(u)$ and $\dfrac{\partial p(u)}{\partial u}$ are defined on staggered grids. All $p(u)$ values can be defined in the half-grid points while all $\dfrac{\partial p(u)}{\partial u}$ values can be defined in the integer-grid points, or vice versa. It is easy to find out that $\dfrac{\partial^2 p(u)}{\partial u^2}$ are also defined in the same grids as $p(u)$ by using the approximation (A.15) twice.

The FDTD method of more complex systems such as elastic wave equations uses more than one set of staggered grids, as shown in Figure A.1(b) with different colors, to define a variety of variables such as particle velocities and auxiliary variables as well as their partial derivatives.

The efficiency of FDTD can be further improved by parallel computing with Message Passing Interface (MPI), a standard that exchanges data from the memory space of one process to that of another process (running in another processor or another computing node connected by high-speed network) through cooperative operations and has become the industry standard of HPC.

A.4 Absorbing Boundary Conditions

Seismic waves propagate in an unbounded subsurface medium in real seismic surveys. For the sake of computational efficiency and storage, a seismic survey is only simulated in a truncated region. If no special techniques are applied on the boundaries of the simulated region, the FD methods would generate strong reflections which do not physically exist in the real seismic survey. In order to generate accurate seismic data that can account for subsurface features and eliminate artificial boundary reflections, a common method is to enforce the absorbing boundary condition (ABC) [29, 46]. Figure A.2 illustrates a 2D scenario in which the truncated velocity model's left, right, and bottom boundaries are padded with absorbing boundaries of a certain thickness. The top of the simulated region is considered to be the free surface of the earth without an ABC being applied. An ABC attenuates the wave amplitudes in those absorbing boundaries to zero and keep the wavefield unaltered outside of the absorbing boundaries.

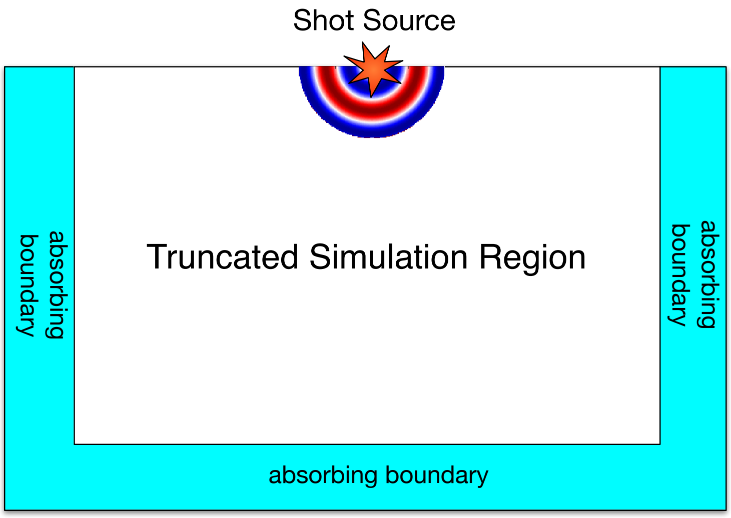


Figure A.2: Absorbing boundaries of a 2D simulation region

The Perfectly Matched Layer (PML) method, which was originally derived for the simulation of electromagnetism with Maxwell equations [9], produces an absorbing boundary layer that can exponentially decay the outgoing waves from the boundary of a truncated simulation region regardless of its incident angle and frequency. For seismic wave simulations, the PML has also been successfully applied in both acoustic and elastic wave equations [63].

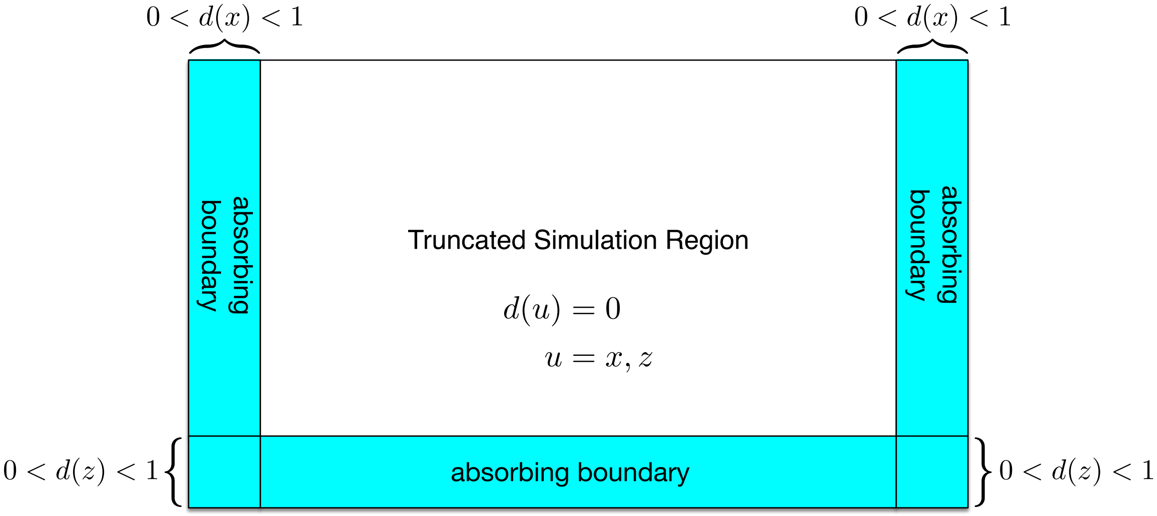


Figure A.3: The damping profile $d(u)$

By defining a damping profile function $d(u)$, $u = x, y, z$, such that $d(u) = 0$ inside the truncated simulation region and $d(u) > 0$ in the PML region, a new complex coordinate $\tilde{u}$ is introduced as

$$\tilde{u}(u) = u + \frac{1}{j\omega}\int_0^u d(s)\mathrm{d}s. \tag{A.16}$$

Then, in wave equations, all partial derivatives with respect to u, i.e. $\dfrac{\partial}{\partial u}$, have to be replaced by

$$\frac{\partial}{\partial \tilde{u}} = \frac{j\omega}{j\omega + d(u)}\frac{\partial}{\partial u}, \tag{A.17}$$

yielding a split-PML scheme as each spatial coordinate $u = x, y, z$ needs to be treated separately.

Three steps are required to apply the PML to a time-domain wave equation. First, the time-domain wave equation is transformed into the frequency domain. For a first-order wave equation with the form

$$\frac{\partial F}{\partial t} = c\frac{\partial G}{\partial u} \tag{A.18}$$

where c is a constant, F and G are arbitrary variables denoting particle velocity or auxiliary variables, the frequency-domain representation is

$$j\omega\hat{F} = c\frac{\partial\hat{G}}{\partial u}. \tag{A.19}$$

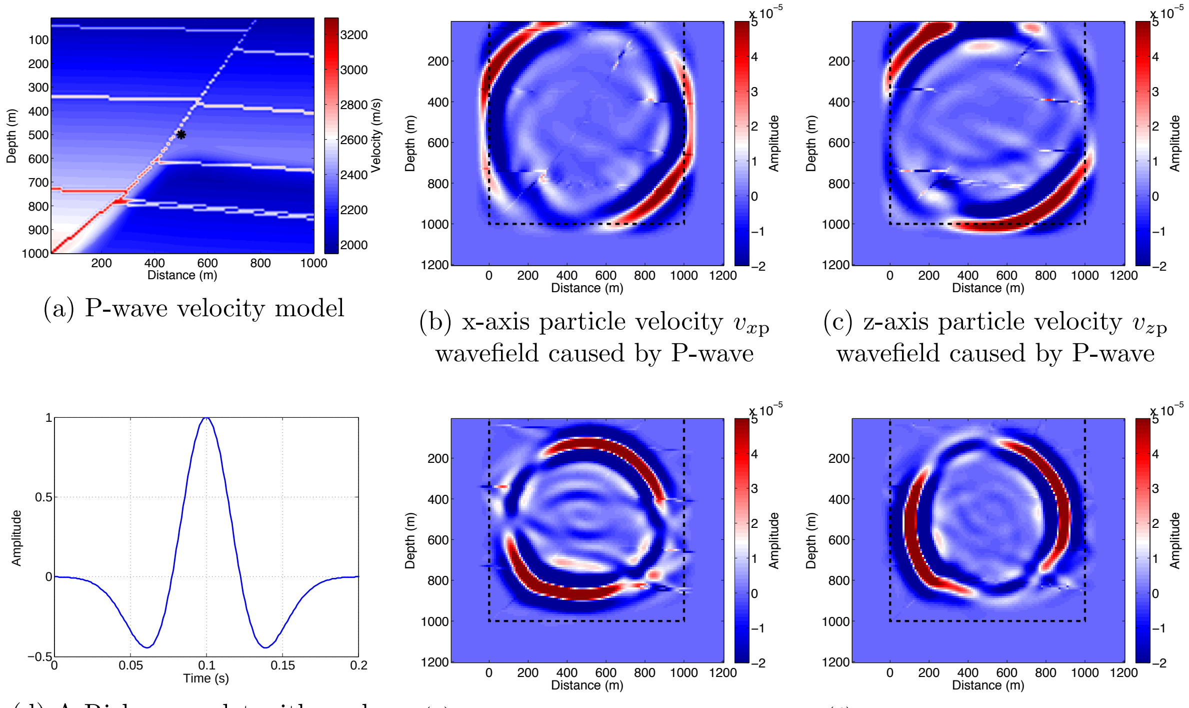
Second, the spatial partial derivative $\dfrac{\partial}{\partial u}$ is replaced by $\dfrac{\partial}{\partial\tilde{u}}$ according to (A.17),

$$j\omega\hat{F} = c\frac{\partial\hat{G}}{\partial\tilde{u}} = \frac{j\omega c}{j\omega + d(u)}\frac{\partial\hat{G}}{\partial u}. \tag{A.20}$$

Third, the frequency-domain equation (A.20) is inverse transformed back to the time domain,

$$\frac{\partial F}{\partial t} + d(u)F = c\frac{\partial G}{\partial u}. \tag{A.21}$$

After these three steps, the FDTD method can be used as before to conduct the numerical simulation.



(a) P-wave velocity model

(b) x-axis particle velocity $v_{x\mathrm{p}}$ wavefield caused by P-wave

(c) z-axis particle velocity $v_{z\mathrm{p}}$ wavefield caused by P-wave

(d) A Ricker wavelet with peak frequency $f_\mathrm{p} = 10\,\mathrm{Hz}$

(e) x-axis particle velocity $v_{x\mathrm{s}}$ wavefield caused by S-wave

(f) z-axis particle velocity $v_{z\mathrm{s}}$ wavefield caused by S-wave

Figure A.4: 2D elastic wavefields generated by FDTD with split-PML

Figure A.4 shows a 2D elastic wave simulation based on a 6th-order staggered-grid FDTD with the split-PML scheme. Figure A.4(a) shows the P-wave velocity model

$v_\mathrm{p}(\mathbf{x})$ with several faults while the S-wave velocity model $v_\mathrm{s}(\mathbf{x}) = v_\mathrm{p}(\mathbf{x})/\sqrt{2}$, and the black asterisk refers to the shot position $\mathbf{x}_s$. Figure A.4(d) plots the source excitation as a Ricker wavelet with peak frequency $f_\mathrm{p} = 10\,\mathrm{Hz}$. Figures A.4(b), A.4(c), A.4(e), A.4(f) depict the particle velocity wavefield snapshots at time $t = 0.33\,\mathrm{s}$ for both the x- and z-axis components caused by the P- and S-waves. The dashed rectangle denotes the boundary between the simulation region and the PML. These figures clearly show that all outgoing waves traveling outside the simulation region have been absorbed in the PML.

Another nonsplit convolutional-PML scheme is introduced in [63] and is used for both FDTD and FDFD methods of this thesis. It transforms (A.17) in the time domain with this form

$$\frac{\partial}{\partial \tilde{u}} = \frac{\partial}{\partial u} + \zeta_u(t) * \frac{\partial}{\partial u} = \frac{\partial}{\partial u} - \left(d(u)H(t)e^{-d(u)t}\right) * \frac{\partial}{\partial u}, \tag{A.22}$$

where $H(t)$ is the Heaviside step function. By defining a new variable $\psi_u^{(n)} \triangleq \zeta_u(t) * \left.\frac{\partial}{\partial u}\right|_{t=n\Delta t}$, the convolution term in (A.22) can be computed as

$$\psi_u^{(n)} = b_u \psi_u^{(n-1)} + (b_u - 1) \left.\frac{\partial}{\partial u}\right|_{t=(n-\frac{1}{2})\Delta t} \tag{A.23}$$

where $b_u = e^{-d(u)\Delta t}$.

A.5 The Finite-difference Frequency-domain (FDFD) Method

The FDFD methods bring in new processing techniques and could offer several advantages over the FDTD counterparts in certain situations. For example, taking the temporal Fourier transform of (A.1) on both sides yields the frequency-domain acoustic wave equation

$$-\frac{\omega^2}{v^2(\mathbf{x})}\hat{p}(\mathbf{x};\omega) - \nabla^2 \hat{p}(\mathbf{x};\omega) = \hat{f}(\mathbf{x};\omega). \tag{A.24}$$

The spatial discretization of (A.24) is the same as the setting for the FDTD method, using either a standard integer-grid scheme or a staggered-grid scheme in addition to the PML.

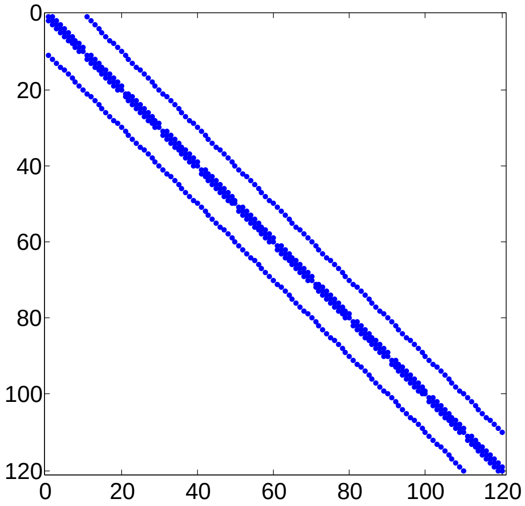


Figure A.5: Visualization of the sparsity pattern of an impedance matrix $\mathbf{B}(\omega)$ of size 120×120 using the first-order FD for a specific frequency ω based on a very tiny model of size 10×12, where blue dots denote its nonzero entries

As an introductory example, by defining

$$\begin{cases} v_{i,j,k} \triangleq v(i\Delta x, j\Delta y, k\Delta z) \\ \hat{p}^{(\omega)}_{i,j,k} \triangleq p(i\Delta x, j\Delta y, k\Delta z; \omega) \\ \hat{f}^{(\omega)}_{i,j,k} \triangleq f(i\Delta x, j\Delta y, k\Delta z; \omega) \end{cases} \tag{A.25}$$

and using the first-order FD, the non-PML region of (A.24) can be discretized for a specific grid point (i, j, k) at frequency ω as

$$\begin{aligned} -\frac{\omega^2}{v^2_{i,j,k}}\hat{p}^{(\omega)}_{i,j,k} &- \left[\frac{\hat{p}^{(\omega)}_{i+1,j,k} - 2\hat{p}^{(\omega)}_{i,j,k} + \hat{p}^{(\omega)}_{i-1,j,k}}{\Delta x^2}\right] \\ &- \left[\frac{\hat{p}^{(\omega)}_{i,j+1,k} - 2\hat{p}^{(\omega)}_{i,j,k} + \hat{p}^{(\omega)}_{i,j-1,k}}{\Delta y^2}\right] \\ &- \left[\frac{\hat{p}^{(\omega)}_{i,j,k+1} - 2\hat{p}^{(\omega)}_{i,j,k} + \hat{p}^{(\omega)}_{i,j,k-1}}{\Delta z^2}\right] = \hat{f}^{(\omega)}_{i,j,k} \end{aligned} \tag{A.26}$$

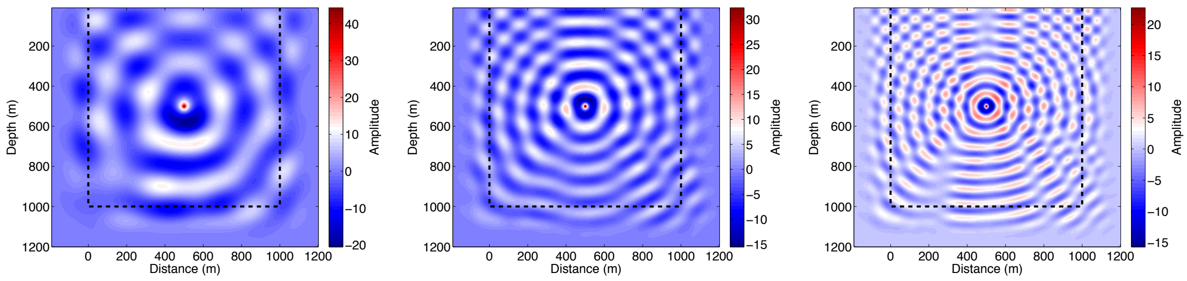
in which $\hat{p}^{(\omega)}_{i,j,k}$ and its 6 direct neighbors form a linear equation. Therefore, a system of linear equations parameterized by ω can be written as

$$\mathbf{B}(\omega)\hat{\mathbf{p}}(\omega) = \hat{\mathbf{f}}(\omega) \tag{A.27}$$

by reshaping $\hat{p}_{i,j,k}^{(\omega)}$ and $\hat{f}_{i,j,k}^{(\omega)}$, $\forall(i,j,k)$, into column vectors $\hat{\mathbf{p}}(\omega)$ and $\hat{\mathbf{f}}(\omega)$, respectively. The square matrix $\mathbf{B}(\omega)$ is the Helmholtz operator matrix, also called the impedance matrix [85], whose coefficients are complex numbers and depend on the frequency, the velocity model, the approximation coefficients and the PML settings. The square matrix $\mathbf{B}(\omega)$ is highly sparse because $\hat{p}_{i,j,k}^{(\omega)}$ is only dependent on its adjacent grid points such that each row of $\mathbf{B}(\omega)$ only contains a few nonzero entries. Figure A.5 visualizes the sparsity pattern of a $\mathbf{B}(\omega)$ of size 120×120 using the first-order FD for a specific frequency ω, based on a very tiny model of size 10×12.

The linear system (A.27) can be solved with a direct-solver method such as inverting $\mathbf{B}(\omega)$ with the LU decomposition [2]. For large-scale simulations, the direct-solver method is no longer affordable because $\mathbf{B}(\omega)$ is so huge that inverting it requires tremendous memory and computational costs. Alternatively, iterative wave-equation solvers [47, 101, 107] are able to accomplish the task with lower memory requirements but higher time costs.

The FDFD methods do not need to compute sequentially since no time steps are involved. Hence one can solve wave equations only in the important part of spectrum. Since the wave equations are independent over shot source positions and frequencies, FDFD can be easily parallelized across them without the need of MPI to achieve a better processing speed, provided that memory requirements are not a significant limitation.



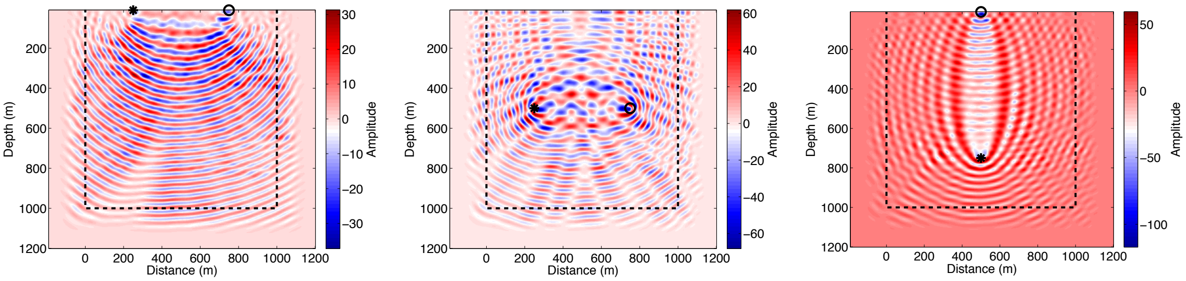
(a) Green's function at 10 Hz (b) Green's function at 20 Hz (c) Green's function at 30 Hz

Figure A.6: Green's functions at different frequencies

The Green's function, which is the impulse response of a PDE, can be easily obtained with FDFD by simply replacing the source term vector $\hat{\mathbf{f}}(\omega)$ in the right-hand side of (A.27) with another column vector $\boldsymbol{\delta}(\mathbf{x}-\mathbf{x}_s)$ that has only one nonzero element whose index corresponds to the source position $\mathbf{x}_s$,

$$\mathbf{B}(\omega)\mathbf{G}(\omega,\mathbf{x}_s) = \boldsymbol{\delta}(\mathbf{x}-\mathbf{x}_s). \tag{A.28}$$

Figure A.6 illustrates several Green's function $\mathbf{G}(\omega,\mathbf{x}_s)$ for the fault model shown in Figure A.4(a), which are calculated at different frequencies ranging from 10 Hz to 30 Hz in parallel. The shot sources are located underground at a depth of 500 m. As before, the boundary between the simulation region and the PML is marked by dashed rectangle and all outgoing waves are absorbed inside the PML.



(a) Surface source/receiver pair

(b) Subsurface source/receiver pair

(c) Subsurface source/surface receiver pair

Figure A.7: Products of two Green's functions show the monochromatic wavepaths for source/receiver pairs, where blue asterisks mark the source locations $\mathbf{x}_s$ and blue circles mark the receiver locations $\mathbf{x}_r$ in the fault model.

The Green's functions are core components for FWI. As was previously seen in Section 3.2, the minimization of the FWI misfit function requires to generate its gradient and Hessian matrix based on the Jacobian matrix, which stacks products of two Green's functions in a row-wise manner. One Green's function corresponds to a source location $\mathbf{x}_s$ and another one corresponds to a receiver location $\mathbf{x}_r$. The product of two Green's functions corresponding to a pair of source and receiver is also referred to as a wavepath. Figure A.7 shows several monochromatic wavepaths for different source/receiver pairs. For a seismic survey using N_ω frequencies, N_s sources and N_r receivers, a total of $N_\omega(N_s + N_r)$ Green's functions need to be evaluated with FDFD to calculate the gradient and Hessian matrix for one FWI iteration, which could result in expensive computational costs. This is the reason that HPC is extensively used for FWI in the study of exploration geophysics.

On the other hand, it is far more difficult to compute Green's functions with FDTD because the time-domain impulse function $\delta(t)\delta(\mathbf{x}-\mathbf{x}_s)$ has an infinitely wide spectrum. In order to obtain accurate FDTD results, the grid point spacing and time step would need to be infinitely small to avoid grid dispersion and numerical instability, which is going to be introduced next.

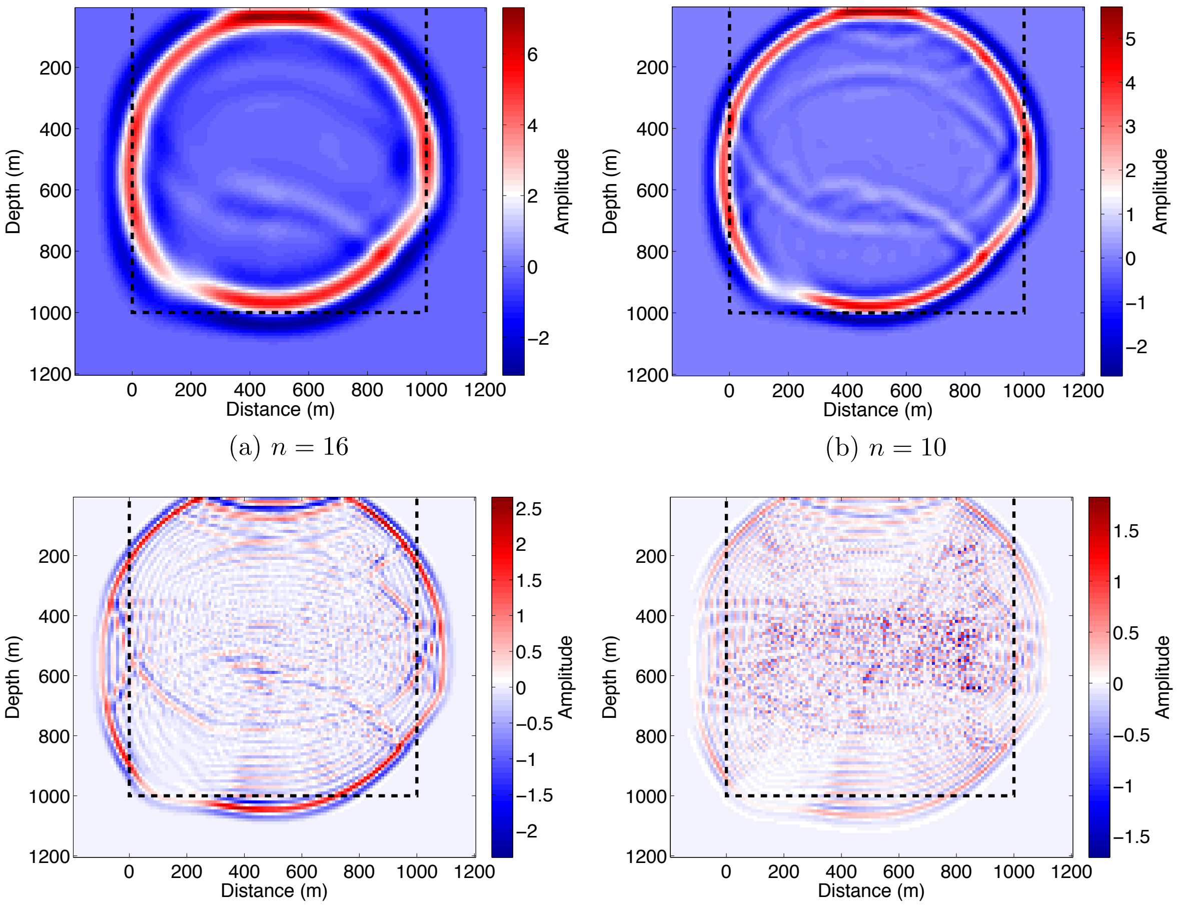
A.6 Grid Dispersion and Instability

The above introduction conveys the message that a variety of parameters need to be determined for seismic wave simulations, such as grid point spacing, source function spectrum, sampling rate, and time step, etc. For the sake of stability and accuracy of the numerical scheme, some prior conditions should be honored when adjusting parameters.

The Nyquist sampling criterion suggests that a sinusoid can be perfectly represented by at least $n = 2$ samples per wavelength. However, the Nyquist rate of $n = 2$ samples per wavelength is far from sufficient for FD methods because it leads to inaccurate estimation of first and second derivatives. As a result, the high-frequency wave components with short wavelengths will slow down and even stop propagating, yielding the numerical artifacts known as grid dispersion.

To avoid the occurrence of grid dispersion, the grid point spacing Δu needs to fulfill the following criterion

$$\Delta u \leq \frac{\lambda_{\min}}{n} = \frac{v_{\min}}{n f_{\max}} \tag{A.29}$$



(a) $n = 16$

(b) $n = 10$

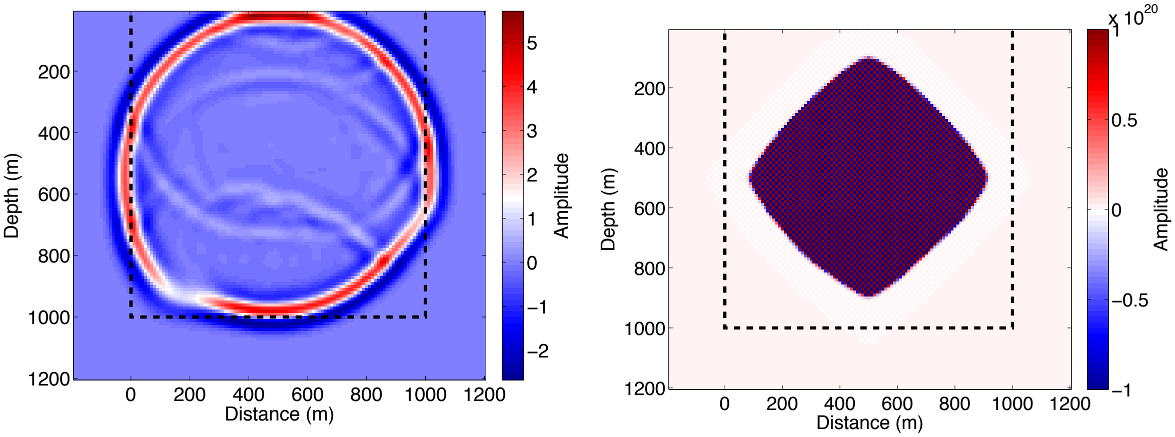
(c) $n = 4$

(d) $n = 2$

Figure A.8: Illustration of grid dispersion

where $\lambda_{\min}$, $v_{\min}$ are the minimal wavelength and velocity, respectively, n is the number of sampling points per minimal wavelength, and $f_{\max}$ is the maximal frequency of the source function spectrum. For reliable simulations, [66, 70, 98] suggest using $n \geq 10$, i.e., the minimum wavelength should cover at least 10 grid points.

Figure A.8 illustrates the effects of grid dispersion on the wavefields, which are generated in the same simulation region with a fixed grid point spacing $\Delta x = \Delta z = 10\,\mathrm{m}$. Several Ricker wavelet functions with increasing peak frequencies f_{p} are used as sources. When a low frequency f_{p} is selected such that $n \geq 10$ samples are used to represent the minimum wavelength, the wavefields are sharply depicted in Figures A.8(a) and A.8(b). For an increased f_{p} with $n = 4$, then slight grid dispersion occurs as shown in Figure A.8(c). The effect of grid dispersion becomes obvious in Figure A.8(d) when only the Nyquist rate $n = 2$ is used.



(a) $\Delta t = 0.001\,\mathrm{s}$

(b) $\Delta t = 0.01\,\mathrm{s}$

Figure A.9: Illustration of instability

Similarly, in order to keep wave simulations stable, the temporal discretization has to satisfy a sampling criterion such that the traveling distance of waves in a time step Δt must be no larger than the grid point spacing, i.e.,

$$\begin{aligned} &\text{2D case}: && \sqrt{2} \cdot v_{\mathrm{p,max}} \cdot \Delta t \leq \min\{\Delta x, \Delta z\} \\ &\text{3D case}: && \sqrt{3} \cdot v_{\mathrm{p,max}} \cdot \Delta t \leq \min\{\Delta x, \Delta y, \Delta z\} \end{aligned} \tag{A.30}$$

where $v_{\mathrm{p,max}}$ is the maximum P-wave velocity. The criterion (A.30) is called Courant-Friedrichs-Lewy (CFL) condition [30]. Figure A.9 illustrates two simulation cases in which the left one satisfies the CFL condition with a sufficiently small Δt and guarantees a stable wave propagation, while the right one violates it with a large Δt and results in a totally unstable wave propagation with infinite amplitudes.

APPENDIX B

THE BORN APPROXIMATION

The Born approximation was proposed by Max Born [15] for scattering theory in quantum physics and has been widely used in different areas. In the context of seismic waves, when the velocity model is changed a little, it is reasonable to suppose that the resulting wavefield would not change substantially, e.g., the scattering waves as reflections that result from a rough perturbation to a smooth background velocity model. The Born approximation provides a linear and invertible relationship between the small model perturbation and the corresponding small wavefield change. Therefore, it has become the basis of most inversion methods based on linearization.

As an example, this chapter derives the Born approximation of the seismic wave equation based on the constant-density acoustic wave equation

$$m(\mathbf{x})\frac{\partial^2 p(\mathbf{x},t)}{\partial t^2} - \nabla^2 p(\mathbf{x},t) = f(\mathbf{x},t) \tag{B.1}$$

where $m(\mathbf{x}) \triangleq \dfrac{1}{v^2(\mathbf{x})}$ is an arbitrary velocity model and $p(\mathbf{x},t)$ is the pressure wavefield.

If an incident model $m(\mathbf{x})$ is disturbed by a small perturbation $\delta m(\mathbf{x})$:

$$m'(\mathbf{x}) \triangleq m(\mathbf{x}) + \delta m(\mathbf{x}), \tag{B.2}$$

the total pressure wavefield $p'(\mathbf{x},t)$ generated by the same excitation that satisfies

$$m'(\mathbf{x})\frac{\partial^2 p'(\mathbf{x},t)}{\partial t^2} - \nabla^2 p'(\mathbf{x},t) = f(\mathbf{x},t) \tag{B.3}$$

can be explained as the summation of the incident pressure wavefield $p(\mathbf{x},t)$ and the wavefield perturbation $\delta p(\mathbf{x},t)$:

$$p'(\mathbf{x},t) \triangleq p(\mathbf{x},t) + \delta p(\mathbf{x},t). \tag{B.4}$$

Subtracting (B.1) from (B.3) leads to the following equation

$$m(\mathbf{x})\frac{\partial^2 \delta p(\mathbf{x},t)}{\partial t^2} - \nabla^2 \delta p(\mathbf{x},t) = -\delta m(\mathbf{x})\frac{\partial^2 p'}{\partial t^2}(\mathbf{x},t). \tag{B.5}$$

This equation cannot be solved yet since its right-hand side still depends on the unknown $\delta p(\mathbf{x},t)$ through $p'(\mathbf{x},t)$. Nevertheless, $\delta p(\mathbf{x},t)$ can be expressed as the following temporal-spatial integration

$$\delta p(\mathbf{x},t) = -\int_0^t \int_{\mathbb{R}^3} G(\mathbf{x},t-\tau;\boldsymbol{\xi})\delta m(\boldsymbol{\xi})\frac{\partial^2 p'}{\partial t^2}(\boldsymbol{\xi},\tau)\mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau \tag{B.6}$$

using the time-domain Green's function $G(\mathbf{x},t-\tau;\boldsymbol{\xi})$ whose source is located at $\boldsymbol{\xi}$. Mathematically, let $\boldsymbol{\mathcal{G}}$ denote a temporal-spatial integral transform with kernel $G(\mathbf{x},t-\tau;\boldsymbol{\xi})$:

$$(\boldsymbol{\mathcal{G}}f)(\mathbf{x},t) \triangleq -\int_0^t \int_{\mathbb{R}^3} G(\mathbf{x},t-\tau;\boldsymbol{\xi})f(\boldsymbol{\xi},\tau)\mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau, \tag{B.7}$$

then (B.6) can be compactly written as

$$\delta\mathbf{p} = -\boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2\mathbf{p}'}{\partial t^2}, \tag{B.8}$$

and the total wavefield $\mathbf{p}'$ including the wavefield perturbation $\delta\mathbf{p}$ becomes

$$\mathbf{p}' = \mathbf{p} - \boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2\mathbf{p}'}{\partial t^2} \tag{B.9}$$

which is called Lippmann-Schwinger equation in the context of quantum physics [76]. Then equation (B.9) can be reformulated as

$$\mathbf{p}' = \left(\mathbf{I} + \boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\right)^{-1}\mathbf{p}. \tag{B.10}$$

This formulation makes an explicit nonlinear relationship between the total wavefield $\mathbf{p}'$ and the incident wavefield $\mathbf{p}$.

The expression $(\mathbf{I}+\boldsymbol{\mathcal{A}})^{-1}$ for some operator $\boldsymbol{\mathcal{A}}$ can be expanded as a Neumann series

$$(\mathbf{I}+\boldsymbol{\mathcal{A}})^{-1} = \mathbf{I} + \sum_{k=1}^{\infty}(-1)^k\boldsymbol{\mathcal{A}}^k = \mathbf{I} - \boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{A}}^2 - \boldsymbol{\mathcal{A}}^3 + \dots \tag{B.11}$$

given that $\|\boldsymbol{\mathcal{A}}\| < 1$ in some kind of norm. In this case, (B.10) can be expanded as

$$\begin{aligned} \mathbf{p}' &= \left(\mathbf{I} + \boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\right)^{-1}\mathbf{p} \\ &= \mathbf{p} - \boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\mathbf{p} + \left(\boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\right)\left(\boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\mathbf{p}\right) - \dots \end{aligned} \tag{B.12}$$

provided that $\left\|\boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\right\| < 1$ is satisfied, and this expression is called a Born series. Using the definition of $\boldsymbol{\mathcal{G}}$, the second and third terms of (B.12) can be explicitly written as

$$-\boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\mathbf{p} \triangleq -\int_0^t\int_{\mathbb{R}^3} G(\mathbf{x}, t-\tau; \boldsymbol{\xi})\delta m(\boldsymbol{\xi})\frac{\partial^2 p}{\partial t^2}(\boldsymbol{\xi}, \tau)\mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau \tag{B.13}$$

and

$$\begin{aligned} &\left(\boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\right)\left(\boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\mathbf{p}\right) \\ &\triangleq \int_0^t\int_{\mathbb{R}^3} G(\mathbf{x}, t-\tau; \boldsymbol{\xi})\delta m(\boldsymbol{\xi})\frac{\partial^2}{\partial t^2}\left[\int_0^\tau\int_{\mathbb{R}^3} G(\boldsymbol{\xi}, \tau-\mu; \boldsymbol{\upsilon})\delta m(\boldsymbol{\upsilon})\frac{\partial^2 p}{\partial \tau^2}(\boldsymbol{\upsilon}, \mu)\mathrm{d}\boldsymbol{\upsilon}\mathrm{d}\mu\right]\mathrm{d}\boldsymbol{\xi}\mathrm{d}\tau. \end{aligned} \tag{B.14}$$

They refer to single scattering and double scattering, respectively. The physical explanation of the single scattering is: the incident wavefield initializes the wave propagation at time 0, generates scattering waves at location $\boldsymbol{\xi}$ and time τ due to the model perturbation $\delta m(\boldsymbol{\xi})$, and these scattering waves reach location $\mathbf{x}$ at time t. The physical explanation of the double scattering is: the incident wavefield initializes the wave propagation at time 0, generates scattering waves at location $\boldsymbol{\upsilon}$ and time μ due to the model perturbation $\delta m(\boldsymbol{\upsilon})$, then generates scattering waves a second time at location $\boldsymbol{\xi}$ and time τ due to the model perturbation $\delta m(\boldsymbol{\xi})$, and these scattering waves reach location $\mathbf{x}$ at time t. Similarly, higher-order terms in (B.12) represent multiple-time scatterings.

The Born approximation takes the single scattering as the approximation for the wavefield perturbation

$$\delta\mathbf{p} \approx -\boldsymbol{\mathcal{G}}\delta\mathbf{m}\frac{\partial^2}{\partial t^2}\mathbf{p}. \tag{B.15}$$

This approximation can be translated back to a PDE,

$$m(\mathbf{x})\frac{\partial^2 \delta p(\mathbf{x},t)}{\partial t^2} - \nabla^2 \delta p(\mathbf{x},t) = -\delta m(\mathbf{x})\frac{\partial^2 p}{\partial t^2}(\mathbf{x},t). \tag{B.16}$$

Comparing with (B.5), the right-hand side of (B.16) depends on the incident wavefield $p(\mathbf{x},t)$, which can be regarded as fixed for solving the wavefield perturbation $\delta p(\mathbf{x},t)$ since $p(\mathbf{x},t)$ is determined from the incident model $m(\mathbf{x})$ alone. Therefore, a linear relationship between the model perturbation $\delta m(\mathbf{x}$ and the wavefield perturbation $\delta p(\mathbf{x},t)$ is established.

REFERENCES

[1] Aharon, M., Elad, M., and Bruckstein, A., "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *Signal Processing, IEEE Transactions on*, vol. 54, pp. 4311–4322, Nov 2006.

[2] Ajo-Franklin, J. B., "Frequency-domain modeling techniques for the scalar wave equation: an introduction," tech. rep., Earth Resources Laboratory, Dept. of Earth, Atmospheric, and Planetary Sciences, Massachusetts Institute of Technology., 2005.

[3] Aki, K. and Richards, P. G., *Quantitative seismology.* Sausalito, California: University Science Books, 2002. second edition.

[4] Alford, R. M., Kelly, K. R., and Boore, D. M., "Accuracy of finitedifference modeling of the acoustic wave equation," *GEOPHYSICS*, vol. 39, no. 6, pp. 834–842, 1974.

[5] Armijo, L., "Minimization of functions having lipschitz continuous first partial derivatives.," *Pacific J. Math.*, vol. 16, no. 1, pp. 1–3, 1966.

[6] Baysal, E., Kosloff, D. D., and Sherwood, J. W. C., "Reverse time migration," *GEOPHYSICS*, vol. 48, no. 11, pp. 1514–1524, 1983.

[7] Beckouche, S. and Ma, J., "Simultaneous dictionary learning and denoising for seismic data," *Geophysics*, vol. 79, no. 3, pp. A27–A31, 2014.

[8] Ben-Hadj-Ali, H., Operto, S., and Virieux, J., "An efficient frequency-domain full waveform inversion method using simultaneous encoded sources," *GEOPHYSICS*, vol. 76, no. 4, pp. R109–R124, 2011.

[9] Berenger, J.-P., "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, vol. 114, pp. 185–200, Oct. 1994.

[10] Beylkin, G., "Imaging of discontinuities in the inverse scattering problem by inversion of a causal generalized radon transform," *Journal of Mathematical Physics*, vol. 26, no. 1, pp. 99–108, 1985.

[11] Bezdek, J. C. and Hathaway, R. J., "Some notes on alternating optimization," in *Advances in Soft Computing — AFSS 2002* (Pal, N. and Sugeno, M., eds.), vol. 2275 of *Lecture Notes in Computer Science*, pp. 288–300, Springer Berlin Heidelberg, 2002.

[12] Birgin, E., Martínez, J., and Raydan, M., "Spectral projected gradient methods: Review and perspectives," *Journal of Statistical Software*, vol. 60, no. 1, pp. 1–21, 2014.

[13] Birgin, E. G., Martínez, J. M., and Raydan, M., "Nonmonotone spectral projected gradient methods on convex sets," *SIAM J. on Optimization*, vol. 10, pp. 1196–1211, Aug. 1999.

[14] Boonyasiriwat, C. and Schuster, G. T., "3d multisource fullwaveform inversion using dynamic random phase encoding," in *SEG Technical Program Expanded Abstracts 2010*, pp. 1044–1049, 2010.

[15] Born, M., "Quantenmechanik der stoßvorgänge," *Zeitschrift für Physik*, vol. 38, no. 11, pp. 803–827, 1926.

[16] Bousquet, O. and Bottou, L., "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems 20* (Platt, J., Koller, D., Singer, Y., and Roweis, S., eds.), pp. 161–168, Cambridge, MA: MIT Press, 2007.

[17] Brenders, A. J. and Pratt, R. G., "Full waveform tomography for lithospheric imaging: results from a blind test in a realistic crustal model," *Geophysical Journal International*, vol. 168, no. 1, pp. 133–151, 2007.

[18] Bunks, C., Saleck, F. M., Zaleski, S., and Chavent, G., "Multiscale seismic waveform inversion," *GEOPHYSICS*, vol. 60, no. 5, pp. 1457–1473, 1995.

[19] Cai, J.-F., Ji, H., Shen, Z., and Ye, G.-B., "Data-driven tight frame construction and image denoising," *Applied and Computational Harmonic Analysis*, vol. 37, no. 1, pp. 89 – 105, 2014.

[20] Candès, E., Demanet, L., Donoho, D., and Ying, L., "Fast discrete curvelet transforms," *Multiscale Modeling and Simulation*, vol. 5, no. 3, pp. 861–899, 2006.

[21] Candès, E. and Donoho, D., "Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges," tech. rep., 1999.

[22] Candès, E., Romberg, J., and Tao, T., "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, pp. 489–509, Feb 2006.

[23] Candès, E. J. and Guo, F., "New multiscale transforms, minimum total variation synthesis: applications to edge-preserving image reconstruction," *Signal Processing*, vol. 82, pp. 1519–1543, 2002.

[24] Candès, E. J. and Demanet, L., "The curvelet representation of wave propagators is optimally sparse," *Communications on Pure and Applied Mathematics*, vol. 58, no. 11, pp. 1472–1528, 2005.

[25] Candès, E. J. and Donoho, D. L., "New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities," *Communications on Pure and Applied Mathematics*, vol. 57, no. 2, pp. 219–266, 2004.

[26] Castellanos, C., Métivier, L., Operto, S., Brossier, R., and Virieux, J., "Fast full waveform inversion with source encoding and second-order optimization methods," *Geophysical Journal International*, vol. 200, no. 2, pp. 720–744, 2015.

[27] Chanerley, A. A. and Alexander, N. A., "An approach to seismic correction which includes wavelet de-noising," in *Proceedings of the Sixth Conference on Computational Structures Technology*, ICCST '02, (Edinburgh, UK, UK), pp. 107–108, Civil-Comp press, 2002.

[28] Chen, S. S., Donoho, D. L., and Saunders, M. A., "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[29] Clayton, R. and Engquist, B., "Absorbing boundary conditions for acoustic and elastic wave equations," *Bulletin of the Seismological Society of America*, vol. 67, pp. 1529–1540, 1977.

[30] Courant, R., Friedrichs, K., and Lewy, H., "Über die partiellen Differenzengleichungen der mathematischen Physik," *Mathematische Annalen*, vol. 100, no. 1, pp. 32–74, 1928.

[31] Dablain, M. A., "The application of highorder differencing to the scalar wave equation," *GEOPHYSICS*, vol. 51, no. 1, pp. 54–66, 1986.

[32] Daubechies, I., *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, 1992.

[33] Demanet, L. and Ying, L., "Curvelets and wave atoms for mirror-extended images," 2007.

[34] Do, M. N. and Vetterli, M., "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, pp. 2091–2106, Dec 2005.

[35] Do, M. and Vetterli, M., "Contourlets," in *Beyond Wavelets* (Welland, G., ed.), vol. 4, (Amsterdam, The Netherlands), pp. 83–105, Academic Press, 2003.

[36] Do, M. and Vetterli, M., "The finite ridgelet transform for image representation," *Image Processing, IEEE Transactions on*, vol. 12, pp. 16–28, Jan 2003.

[37] Dong, S., Cai, J., Guo, M., Suh, S., Zhang, Z., Wang, B., and Li, Z., "Least-squares reverse time migration: towards true amplitude imaging and improving the resolution," in *SEG Technical Program Expanded Abstracts 2012*, pp. 1–5, 2012.

[38] Donoho, D. L. and Elad, M., "Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell 1$ minimization," *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[39] Donoho, D., "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, pp. 1289–1306, April 2006.

[40] Donoho, D. and Huo, X., "Uncertainty principles and ideal atomic decomposition," *Information Theory, IEEE Transactions on*, vol. 47, pp. 2845–2862, Nov 2001.

[41] Duarte, M. F. and Eldar, Y. C., "Structured compressed sensing: From theory to applications," *IEEE Transactions on Signal Processing*, vol. 59, pp. 4053–4085, Sept 2011.

[42] Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T., "Efficient projections onto the l1-ball for learning in high dimensions," in *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, (New York, NY, USA), pp. 272–279, ACM, 2008.

[43] Duijndam, A. J. W., Schonewille, M. A., and Hindriks, C. O. H., "Reconstruction of bandlimited signals, irregularly sampled along one spatial direction," *GEOPHYSICS*, vol. 64, no. 2, pp. 524–538, 1999.

[44] Elad, M. and Aharon, M., "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, pp. 3736–3745, Dec 2006.

[45] Engan, K., Aase, S., and Hakon Husoy, J., "Method of optimal directions for frame design," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 5, pp. 2443–2446 vol.5, 1999.

[46] Engquist, B. and Majda, A., "Absorbing boundary conditions for numerical simulation of waves," *Proceedings of the Natinoal Academy of Sciences*, vol. 74, pp. 1765–1766, 1977.

[47] Erlangga, Y. A. and Herrmann, F. J., "An iterative multilevel method for computing wavefields in frequencydomain seismic inversion," in *SEG Technical Program Expanded Abstracts 2008*, pp. 1956–1960, 2008.

[48] Etgen, J. and Regone, C., "Strike shooting, dip shooting, widepatch shooting – Does prestack migration care? A model study," in *Expanded Abstracts*, vol. 98, pp. 66–69, 1998.

[49] Fomel, S. and Liu, Y., "Seislet transform and seislet frame," *GEOPHYSICS*, vol. 75, no. 3, pp. V25–V38, 2010.

[50] Gray, S. H., Etgen, J., Dellinger, J., and Whitmore, D., "Seismic migration problems and solutions," *GEOPHYSICS*, vol. 66, no. 5, pp. 1622–1640, 2001.

[51] Gubernatis, J., Domany, E., Krumhansl, J., and Huberman, M., "The Born approximation in the theory of the scattering of elastic waves by flaws," *Journal of Applied Physics*, vol. 48, no. 7, pp. 2812–2819, 1977.

[52] Guitton, A. and Díaz, E., "Attenuating crosstalk noise with simultaneous source full waveform inversion," *Geophysical Prospecting*, vol. 60, no. 4, pp. 759–768, 2012.

[53] Hennenfent, G., Fenelon, L., and Herrmann, F., "Nonequispaced curvelet transform for seismic data reconstruction: A sparsity-promoting approach," *GEOPHYSICS*, vol. 75, no. 6, pp. WB203–WB210, 2010.

[54] Hennenfent, G. and Herrmann, F., "Seismic denoising with nonuniformly sampled curvelets," *Computing in Science Engineering*, vol. 8, pp. 16–25, May 2006.

[55] Hennenfent, G. and Herrmann, F. J., "Simply denoise: Wavefield reconstruction via jittered undersampling," *GEOPHYSICS*, vol. 73, no. 3, pp. V19–V28, 2008.

[56] Herrmann, F. J. and Li, X., "Efficient least-squares imaging with sparsity promotion and compressive sensing," *Geophysical Prospecting*, vol. 60, no. 4, pp. 696–712, 2012.

[57] Herrmann, F. J., Li, X., Aravkin, A. Y., and van Leeuwen, T., "A modified, sparsity-promoting, Gauss-Newton algorithm for seismic waveform inversion," 2011.

[58] Herrmann, F. J., Wang, D., Hennenfent, G., and Moghaddam, P. P., "Curvelet-based seismic data processing: A multiscale and nonlinear approach," *GEOPHYSICS*, vol. 73, no. 1, pp. A1–A5, 2008.

[59] Hood, P., *Developments in Geophysical Methods*, ch. Migration. London: Applied Science, 1981.

[60] Jang, U., Min, D.-J., and Shin, C., "Comparison of scaling methods for waveform inversion," *Geophysical Prospecting*, vol. 57, no. 1, pp. 49–59, 2009.

[61] Kelly, K. R., Ward, R. W., Treitel, S., and Alford, R. M., "Synthetic seismograms: A finite difference approach," *GEOPHYSICS*, vol. 41, no. 1, pp. 2–27, 1976.

[62] Kirolos, S., Laska, J., Wakin, M., Duarte, M., Baron, D., Ragheb, T., Massoud, Y., and Baraniuk, R., "Analog-to-information conversion via random demodulation," in *Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on*, pp. 71–74, Oct 2006.

[63] Komatitsch, D. and Martin, R., "An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation," *GEOPHYSICS*, vol. 72, no. 5, pp. SM155–SM167, 2007.

[64] Krebs, J. R., Anderson, J. E., Hinkley, D., Neelamani, R., Lee, S., Baumstein, A., and Lacasse, M.-D., "Fast full-wavefield seismic inversion using encoded sources," *GEOPHYSICS*, vol. 74, no. 6, pp. WCC177–WCC188, 2009.

[65] Lailly, P., "The seismic inverse problem as a sequence of before stack migrations: Conference on inverse scattering, theory and application," *Conference on Inverse Scattering, Theory and Application, Society for Industrial and Applied Mathematics, Expanded Abstracts*, pp. 206–220, 1983.

[66] Larsen, S. and Schultz, C. A., "ELAS3D: 2D/3D elastic finite difference wave propagation code: Technical Report No. UCRL-MA-121792," tech. rep., 1995.

[67] Laska, J. N., Kirolos, S., Duarte, M. F., Ragheb, T. S., Baraniuk, R. G., and Massoud, Y., "Theory and implementation of an analog-to-information converter using random demodulation," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pp. 1959–1962, May 2007.

[68] Lebrun, M. and Leclaire, A., "An Implementation and Detailed Analysis of the K-SVD Image Denoising Algorithm," *Image Processing On Line*, vol. 2, pp. 96–133, 2012.

[69] Lesage, S., Gribonval, R., Bimbot, F., and Benaroya, L., "Learning unions of orthonormal bases with thresholded singular value decomposition," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 5, pp. v/293–v/296 Vol. 5, March 2005.

[70] Levander, A. R., "Fourthorder finitedifference p-sv seismograms," *GEOPHYSICS*, vol. 53, no. 11, pp. 1425–1436, 1988.

[71] Lewicki, M. S. and Olshausen, B. A., "Probabilistic framework for the adaptation and comparison of image codes," *J. Opt. Soc. Am. A*, vol. 16, pp. 1587–1601, Jul 1999.

[72] Li, F. and Zeng, T., "A universal variational framework for sparsity-based image inpainting," *IEEE Transactions on Image Processing*, vol. 23, pp. 4242–4254, Oct 2014.

[73] Li, X., Aravkin, A. Y., van Leeuwen, T., and Herrmann, F. J., "Fast randomized full-waveform inversion with compressive sensing," *GEOPHYSICS*, vol. 77, no. 3, pp. A13–A17, 2012.

[74] Li, X., Esser, E., and Herrmann, F. J., "Modified gauss-newton full-waveform inversion explained – why sparsity-promoting updates do matter," *To appear in Geophysics*, 2016.

[75] Li, X. and Herrmann, F. J., "Efficient full-waveform inversion with marine acquisition geometry," *GeoConvention 2012:Vision*, 2012.

[76] Lippmann, B. A. and Schwinger, J., "Variational principles for scattering processes. i," *Phys. Rev.*, vol. 79, pp. 469–480, Aug 1950.

[77] Liu, B. and Sacchi, M. D., "Minimum weighted norm interpolation of seismic records," *GEOPHYSICS*, vol. 69, no. 6, pp. 1560–1568, 2004.

[78] Loris, I., Nolet, G., Daubechies, I., and Dahlen, F. A., "Tomographic inversion using ℓ_1-norm regularization of wavelet coefficients," *Geophysical Journal International*, vol. 170, no. 1, pp. 359–370, 2007.

[79] Luo, C., *Non-uniform sampling: algorithms and architectures.* PhD thesis, Georgia Institute of Technology, 2012.

[80] Lustig, M., Donoho, D., and Pauly, J. M., "Sparse mri: The application of compressed sensing for rapid mr imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[81] Madagascar Development Team, *Madagascar Software, Version 1.6.5.* http://www.ahay.org/, 2014.

[82] Mairal, J., Elad, M., and Sapiro, G., "Sparse representation for color image restoration," *Image Processing, IEEE Transactions on*, vol. 17, pp. 53–69, Jan 2008.

[83] Mallat, S. and Zhang, Z., "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, pp. 3397–3415, Dec 1993.

[84] Mallat, S., *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way.* Academic Press, 3rd ed., 2008.

[85] Marfurt, K. J., "Accuracy of finitedifference and finiteelement modeling of the scalar and elastic wave equations," *GEOPHYSICS*, vol. 49, no. 5, pp. 533–549, 1984.

[86] McMechan, G. A., "Migration by extrapolation of time-dependent boundary values*," *Geophysical Prospecting*, vol. 31, no. 3, pp. 413–420, 1983.

[87] Moghaddam, P. P. and Herrmann, F. J., “Randomized fullwaveform inversion: a dimenstionalityreduction approach,” in *SEG Technical Program Expanded Abstracts 2010*, pp. 977–982, 2010.

[88] Montanari, A., “Graphical models concepts in compressed sensing,” in *Compressed Sensing* (Eldar, Y. C. and Kutyniok, G., eds.), pp. 394–438, Cambridge University Press, 2012. Cambridge Books Online.

[89] Naghizadeh, M. and Sacchi, M. D., “Beyond alias hierarchical scale curvelet interpolation of regularly and irregularly sampled seismic data,” *GEOPHYSICS*, vol. 75, no. 6, pp. WB189–WB202, 2010.

[90] Natarajan, B. K., “Sparse approximate solutions to linear systems,” *SIAM J. Comput.*, vol. 24, pp. 227–234, Apr. 1995.

[91] Nati, N. S. and Jaakkola, T., “Weighted low-rank approximations,” in *In 20th International Conference on Machine Learning*, pp. 720–727, AAAI Press, 2003.

[92] Neelamani, R., Baumstein, A. I., Gillard, D. G., Hadidi, M. T., and Soroka, W. L., “Coherent and random noise attenuation using the curvelet transform,” *The Leading Edge*, vol. 27, no. 2, pp. 240–248, 2008.

[93] Nocedal, J., “Updating quasi-Newton matrices with limited storage,” *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[94] Nocedal, J. and Wright, S., *Numerical optimization.* Springer Science & Business Media, 2006.

[95] Olshausen, B. A. and Field, D. J., “Natural image statistics and efficient coding,” *Network: Computation in Neural Systems*, vol. 7, no. 2, pp. 333–339, 1996. PMID: 16754394.

[96] Olshausen, B. A. and Field, D. J., “Sparse coding with an overcomplete basis set: A strategy employed by v1?,” *Vision Research*, vol. 37, no. 23, pp. 3311 – 3325, 1997.

[97] Pan, W., Innanen, K. A., Margrave, G. F., and Cao, D., “Efficient pseudo-Gauss-Newton full-waveform inversion in the τ-p domain,” *GEOPHYSICS*, vol. 80, no. 5, pp. R225–R14, 2015.

[98] Panning, M., Dreger, D., and Tkalčić, H., “Near-source velocity structure and isotropic moment tensors: A case study of the long valley caldera,” *Geophysical Research Letters*, vol. 28, no. 9, pp. 1815–1818, 2001.

[99] Pati, Y., Rezaiifar, R., and Krishnaprasad, P., “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pp. 40–44 vol.1, Nov 1993.

[100] Peleg, T. and Elad, M., "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Transactions on Image Processing*, vol. 23, pp. 2569–2582, June 2014.

[101] Plessix, R.-E., "A helmholtz iterative solver for 3d seismic-imaging problems," *GEOPHYSICS*, vol. 72, no. 5, pp. SM185–SM194, 2007.

[102] Plessix, R.-E. and Mulder, W. A., "Frequency-domain finite-difference amplitude-preserving migration," *Geophysical Journal International*, vol. 157, no. 3, pp. 975–987, 2004.

[103] Pratt, R. G., "Frequency-domain elastic wave modeling by finite differences: A tool for crosshole seismic imaging," *GEOPHYSICS*, vol. 55, no. 5, pp. 626–632, 1990.

[104] Pratt, R. G., "Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model," *GEOPHYSICS*, vol. 64, no. 3, pp. 888–901, 1999.

[105] Pratt, R. G., Shin, C., and Hick, G. J., "Gauss-Newton and full Newton methods in frequency–space seismic waveform inversion," *Geophysical Journal International*, vol. 133, no. 2, pp. 341–362, 1998.

[106] Ren, H., Wang, H., and Chen, S., "Least-squares reverse time migration in frequency domain using the adjoint-state method," *Journal of Geophysics and Engineering*, vol. 10, no. 3, p. 035002, 2013.

[107] Riyanti, C. D., Erlangga, Y. A., Plessix, R.-E., Mulder, W. A., Vuik, C., and Oosterlee, C., "A new iterative solver for the time-harmonic wave equation," *GEOPHYSICS*, vol. 71, no. 5, pp. E57–E63, 2006.

[108] Romberg, J., "Compressive sensing by random convolution," *SIAM Journal on Imaging Sciences*, vol. 2, no. 4, pp. 1098–1128, 2009.

[109] Romero, L. A., Ghiglia, D. C., Ober, C. C., and Morton, S. A., "Phase encoding of shot records in prestack migration," *GEOPHYSICS*, vol. 65, no. 2, pp. 426–436, 2000.

[110] Ronen, J., "Waveequation trace interpolation," *GEOPHYSICS*, vol. 52, no. 7, pp. 973–984, 1987.

[111] Rubinstein, R., Zibulevsky, M., and Elad, M., "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *Signal Processing, IEEE Transactions on*, vol. 58, pp. 1553–1564, March 2010.

[112] Schmidt, M. W., Berg, E., Friedlander, M. P., and Murphy, K. P., "Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm," in *International Conference on Artificial Intelligence and Statistics*, pp. 456–463, April 2009.

[113] Schönemann, P., "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[114] Schuster, G. T., "Leastsquares crosswell migration," in *SEG Technical Program Expanded Abstracts 1993*, pp. 110–113, 1993.

[115] Sezer, O., Guleryuz, O., and Altunbasak, Y., "Approximation and compression with sparse orthonormal transforms," *Image Processing, IEEE Transactions on*, vol. 24, pp. 2328–2343, Aug 2015.

[116] Sezer, O. G., *Data-driven transform optimization for next generation multimedia applications*. PhD thesis, Georgia Institute of Technology, 2012.

[117] Shahidi, R., Tang, G., Ma, J., and Herrmann, F. J., "Application of randomized sampling schemes to curvelet-based sparsity-promoting seismic data recovery," *Geophysical Prospecting*, vol. 61, no. 5, pp. 973–997, 2013.

[118] Shan, H., Ma, J., and Yang, H., "Comparisons of wavelets, contourlets and curvelets in seismic denoising," *Journal of Applied Geophysics*, vol. 69, no. 2, pp. 103 – 115, 2009.

[119] Shen, Y., Li, J., Zhu, Z., Cao, W., and Song, Y., "Image reconstruction algorithm from compressed sensing measurements by dictionary learning," *Neurocomputing*, vol. 151, Part 3, pp. 1153 – 1162, 2015.

[120] Shin, C., Jang, S., and Min, D.-J., "Improved amplitude preservation for prestack depth migration by inverse scattering theory," *Geophysical Prospecting*, vol. 49, no. 5, pp. 592–606, 2001.

[121] Sirgue, L. and Pratt, R. G., "Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies," *GEOPHYSICS*, vol. 69, no. 1, pp. 231–248, 2004.

[122] Starck, J.-L., Murtagh, F., and Fadili, J., *Sparse Image and Signal Processing: Wavelets and Related Geometric Multiscale Analysis*. Cambridge University Press, 2015.

[123] Symes, W., "Source synthesis for waveform inversion," in *SEG Technical Program Expanded Abstracts 2010*, pp. 1018–1022, 2010.

[124] Symes, W. W., "Migration velocity analysis and waveform inversion," *Geophysical Prospecting*, vol. 56, no. 6, pp. 765–790, 2008.

[125] Tan, S. and Huang, L., "Least-squares reverse-time migration with a wavefield-separation imaging condition and updated source wavefields," *GEOPHYSICS*, vol. 79, no. 5, pp. S195–S205, 2014.

[126] Tang, G., Ma, J.-W., and Yang, H.-Z., "Seismic data denoising based on learning-type overcomplete dictionaries," *Applied Geophysics*, vol. 9, no. 1, pp. 27–32, 2012.

[127] Tarantola, A., "Inversion of seismic reflection data in the acoustic approximation," *GEOPHYSICS*, vol. 49, no. 8, pp. 1259–1266, 1984.

[128] Tarantola, A., "A strategy for nonlinear elastic inversion of seismic reflection data," *GEOPHYSICS*, vol. 51, no. 10, pp. 1893–1903, 1986.

[129] Tarantola, A., *Inverse problem theory: Methods for data fitting and parameter estimation.* Elsevier, Amsterdam, 1987.

[130] Tibshirani, R., "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[131] Tibshirani, R., "Regression shrinkage and selection via the LASSO: a retrospective," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 3, pp. 273–282, 2011.

[132] Tropp, J. and Gilbert, A., "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, pp. 4655–4666, Dec 2007.

[133] van den Berg, E. and Friedlander, M. P., "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2009.

[134] Vidal, R., Ma, Y., and Sastry, S., "Generalized principal component analysis (gpca)," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, pp. 1945–1959, Dec 2005.

[135] Virieux, J. and Operto, S., "An overview of full-waveform inversion in exploration geophysics," *GEOPHYSICS*, vol. 74, no. 6, pp. WCC1–WCC26, 2009.

[136] Wang, B., Wu, R.-S., Chen, X., and Li, J., "Simultaneous seismic data interpolation and denoising with a new adaptive method based on dreamlet transform," *Geophysical Journal International*, vol. 201, no. 2, pp. 1182–1194, 2015.

[137] Warner, M., Ratcliffe, A., Nangoo, T., Morgan, J., Umpleby, A., Shah, N., Vinje, V., Štekl, I., Guasch, L., Win, C., Conroy, G., and Bertrand, A., "Anisotropic 3d full-waveform inversion," *GEOPHYSICS*, vol. 78, no. 2, pp. R59–R80, 2013.

[138] Whitmore, N. D., "Iterative depth migration by backward time propagation," in *SEG Technical Program Expanded Abstracts 1983*, pp. 382–385, 1983.

[139] Wu, R. and Aki, K., "Scattering characteristics of elastic waves by an elastic heterogeneity," *GEOPHYSICS*, vol. 50, no. 4, pp. 582–595, 1985.

[140] Xue, Z. and Zhu, H., "Full waveform inversion with sparsity constraint in seislet domain," in *SEG Technical Program Expanded Abstracts 2015*, pp. 1382–1387, 2015.

[141] Yang, J., Wright, J., Huang, T., and Ma, Y., "Image super-resolution via sparse representation," *Image Processing, IEEE Transactions on*, vol. 19, pp. 2861–2873, Nov 2010.

[142] Yu, S., Ma, J., Zhang, X., and Sacchi, M. D., "Interpolation and denoising of high-dimensional seismic data by learning a tight frame," *GEOPHYSICS*, vol. 80, no. 5, pp. V119–V132, 2015.

[143] Zhang, R. and Ulrych, T., "Physical wavelet frame denoising," *GEOPHYSICS*, vol. 68, no. 1, pp. 225–231, 2003.

[144] Zhu, L., Liu, E., and McClellan, J. H., "Fast online orthonormal dictionary learning for efficient full waveform inversion," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1427–1431, March 2016.

[145] Zhu, L., Liu, E., and McClellan, J. H., "Seismic data denoising through multiscale and sparsity-promoting dictionary learning," *GEOPHYSICS*, vol. 80, no. 6, pp. WD45–WD57, 2015.

[146] Zwartjes, P. M. and Sacchi, M. D., "Fourier reconstruction of nonuniformly sampled, aliased seismic data," *GEOPHYSICS*, vol. 72, no. 1, pp. V21–V32, 2007.