Large-scale seismic data compression: application to full waveform inversion and extended image volume

by

Yiming Zhang

B.Sc. Geophysics, China University of Petroleum, 2015

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

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Geophysics)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

April 2018

Abstract

Conventional oil and gas fields are increasingly difficult to explore and image, resulting in a call for more complex wave-equation based inversion algorithms that require dense long-offset samplings. Consequently, there is an exponential growth in the size of data volumes and prohibitive demands on computational resources. In this work, we propose a method to compress and process seismic data directly in a low-rank tensor format, which drastically reduces the amount of storage required to represent the data. We first outline how seismic data exhibits low-rank structure in a particular transform-domain, which can be exploited to compress the dense data in one extremely storage-efficient tensor format when the data is fully sampled. In the more realistic case of missing data, we can use interpolation techniques based on the same tensor format to recover fully sampled data volume in compressed form. In either case, once we have our data represented in its compressed tensor form, we design an algorithm to extract source or receiver gathers directly from the compressed parameters. This extraction process can be done on-the-fly directly on the compressed data, in the full waveform inversion context, and does not require scanning through the entire dataset in order to form shot gathers. To the best of our knowledge, this work is one of the first major contributions to working with seismic data applications directly in the compressed domain without reconstructing the entire data volume. We use a stochastic inversion approach, which works with small subsets of source experiments at each iteration, further to reduce the computational and memory costs of full waveform inversion. We also demonstrate how this data compression and extraction technique can be applied to forming full subsurface image gathers through probing techniques.

Lay Summary

In this thesis, we have developed the computationally efficient framework to directly process greatly-compressed seismic data in the subsequent processing procedures. The proposed technique is built upon the low-rank structure of seismic data in some specific domain, we therefore enable an extremely storage tensor format to well represent our large-scale seismic data sets in much smaller sized parameters. In addition, our framework can work well irrespective of our sampling regime. The codes of our approach can be easily embedded into other processing frameworks and utilizing these parameters in compressed form substantially reduce our memory costs without degrading subsequent results.

Preface

All the presented work in this thesis is carried out under the supervision of Dr. Felix Herrmann in the Seismic Laboratory for Imaging and Modelling (SLIM) at the University of British Columbia.

The entire algorithms and applications are mainly prepared with MATLAB and its parallel computing toolbox (https://www.mathworks.com/). The inversion framework used in this work was initially written by Tristan van Leeuwen, and modified, developed by Curt Da Silva, which become a public software called WAVEFORM (https://github.com/slimgroup/WAVEFORM). Some other open source codes utilized in this thesis include the linear operator toolbox SPOT (http://www.cs.ubc.ca/labs/scl/spot/) and Hierarchical Tucker toolbox htucker (http://anchp.epfl.ch/htucker). The implementation of the HT interpolation algorithm mentioned in Chapter 3 was originally proposed by Curt Da Silva.

A version of Chapter 2 and partial work in Chapter 3 has been published as an extended abstract and was an oral presentation at the 2017 Society of Exploration Geophysicists (SEG) annual meeting in Houston. A version and Chapter 2, 3 and 4 was presented in the SINBAD 2017 fall consortium meeting.

> Yiming Zhang, Curt Da Silva, Rajiv Kumar, and Felix J. Herrmann, Massive 3D seismic data compression and inversion with hierarchical Tucker, *SEG Technical Program Expanded Abstracts, 2017*.

> Yiming Zhang, Curt Da Silva, Rajiv Kumar, and Felix J. Herrmann, Massive seismic data compression & recovery w/ on-the-fly data extraction, *SINBAD Fall consortium, 2017*.

Theoretical section about extraction technique together with most of

the numerical results in the main bady has been submitted to Geophyscis on Feb. 9, 2018 with Curt Da Silva, Rajiv Kumar and my supervisor Felix Herrmann. I, along with Curt Da Silva, was in charge of the algorithms development and test. The experiments design, in the case study of full waveform inversion, was extensively discussed with Curt Da Silva, Rajiv Kumar and Felix Herrmann. Rajiv Kumar also gave me a lot of insightful suggestions and helpful guide to the experiment setting up in the case study of extended image volumes. I was responsible for the coding, conducting the numerical experiments, producing the figures, tables as well as software release related to this work. All the authors were involved in the initial layout of this work. I, along with Curt Da Silva, wrote the manuscript jointly. Felix Herrmann contributed to some editorial corrections. Curt Da Silva and I made an equal contribution (approximately 50%) to developing this research idea applied to seismic applications.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acknowledgements

My sincere gratitude first goes to my advisor Dr. Felix Herrmann for his generous support and valuable inputs for my research. His enthusiasm in creative ideas as well as critical thinking skills profoundly influence me throughout all my studies at UBC and I will definitely benefit from them in my entire life.

I would like to express my appreciation to Miranda Joyce for all her kind help. She, as a program coordinator, is the one who can always prepare for everything you will need before you really need them. In addition, I am also thankful to Henryk Modzelewski for his administration and maintenance of softwares environment used by the SLIM group. Also, I would like to thank my beautiful girlfriend Qin for her support and encouragement throughout my time at UBC.

Last but certainly not least, many thanks to all the members in SLIM group. In particular, I appreciate the time spending and useful discussions with Rajiv Kumar, Curt Da Silva for the proof-reading of the manuscript of my thesis, and the forever friendship with Zhilong Fang, Mengmeng Yang, Dr. Rongrong Wang, Dr. Rajiv Kumar, Ali Alfaraj, Shashin Sharan, Ali Siahkoohi.

This research was carried out as part of the SINBAD II project with the support of the member organizations of the SINBAD Consortium. The authors wish to acknowledge the SENAI CIMATEC Supercomputing Center for Industrial Innovation, with support from BG Brasil, Shell, and the Brazilian Authority for Oil, Gas and Biofuels (ANP), for the provision and operation of computational facilities and the commitment to invest in Research & Development.

Dedication

To my parents,
my girlfriend Qin

Chapter 1

Introduction

1.1 Seismic exploration

Geophysical exploration aims to determine the potential locations and sizes of oil and gas reservoirs, by elucidating the geological structure of the subsurface and the distribution of stratigraphy under investigation. In cooperation with petroleum geologists and reservoir engineers, geophysicists achieve this objective by means of mapping the geophysical data acquired from the surface to underlying structures and properties. Various geophysical methods exist in the application of data, ranging from seismic, gravitational, electrical, magnetic and electromagnetic, among which seismic techniques have been extensively applied to oil and gas exploration activities all over the world. These techniques are linked to the capability of deeper penetration and small anomaly detection with higher resolution (Sheriff and Geldart (1995); Yilmaz (2001)). As such, industries have successfully enabled seismic methods to assess the likelihood of hydrocarbon accumulations and further locate the prospective drilling targets over past decades.

In seismic experiments, dynamite explosions or vibroseis vehicles on land, or airguns for offshore exploration are employed as a controlled *source* of acoustic or elastic energy, which propagates throughout the different earth strata. Some portions of the energy will reflect off the interfaces between two stratigraphies with a velocity and/or density contrast. *Receivers*, such as geophones or hydrophones, placed at discrete positions at the surface, record reflected energy over a specified period of time. As illustrated in Figure 1.1, a seismic *trace* presents a time-series response to a source at one receiver location, measuring ground motions (land) or wave pressure (sea). For a fixed source, a conventional *common shot or source gather* is the side-by-side display of a group of traces from all the receivers corresponding to this single source (Figure 1.1). Similarly, it is straightforward to form the *common receiver gather* corresponding to all the shots recorded

at a single receiver position. Some other classes of gathers exist, as seismic traces share the same geometry or seismic attributes, such as offset, azimuth, incidence angle, etc. Although those types of gathers may play important roles for different purposes in seismic processing, we concentrate on the methods related to common shot/receiver gathers in this thesis, thus referring interested readers to Yilmaz (2001) for detailed explanations. To describe the seismic shot-based processing algorithms in this work, we adopt Berkhouts monochromatic *data matrix* notation (Berkhout (2012)) in which each column stands for a monochromatic common shot gather and each row represents a monochromatic common receiver gather, as shown in Figure 1.2. This Berkhouts monochromatic *data matrix* notation is used for a 2D seismic line, and it can extend to 3D seismic surveys in the canonical organization (source x, source y, receiver x, receiver y) rather than the noncanonical organization (source x, receiver x, source y, receiver y). It's crucial to keep this permutation of the data matrix in mind, which we explore further in the main body of this thesis. As seismic acquisition can be carried out along a line or over an area at the surface to delineate a 2D vertical profile underneath or 3D subsurface volume, the recording seismic data becomes a cube in a 2D seismic survey while it leads to five dimensions (two source coordinates, two receiver coordinates, time) of the data volume for a 3D seismic experiment. These massive seismic data, carrying information about underlying media heterogeneities, are used to invert subsurface images as well as physical properties in downstream processes. Next, interpretations of previously processed data are critical in building a reliable model of subsurface geology and ultimately identifying hydrocarbon deposits.

1.2 Motivation

Recent years have seen a surge of seismic activities moving towards challenge-to-image areas, which require higher resolution and more accurate placement of the reservoir targets in increasingly complex regions, such as sub-salt or thrust belts (Leveille et al. (2011); Jones and Davison (2014)). As a result, industries typically not only acquire finely-sampled data under Nyquist's sampling criterion to avoid aliasing in downstream steps (Abma et al. (2007); Gray (2013)), but they may also need additional azimuths and long-offset data to improve imaging with better illumination of deeper geologies and more subsurface angles (Virieux and Operto (2009); Mothi et al. (2013)). Nevertheless, given the large scale and high dimensionality of 3D seismic

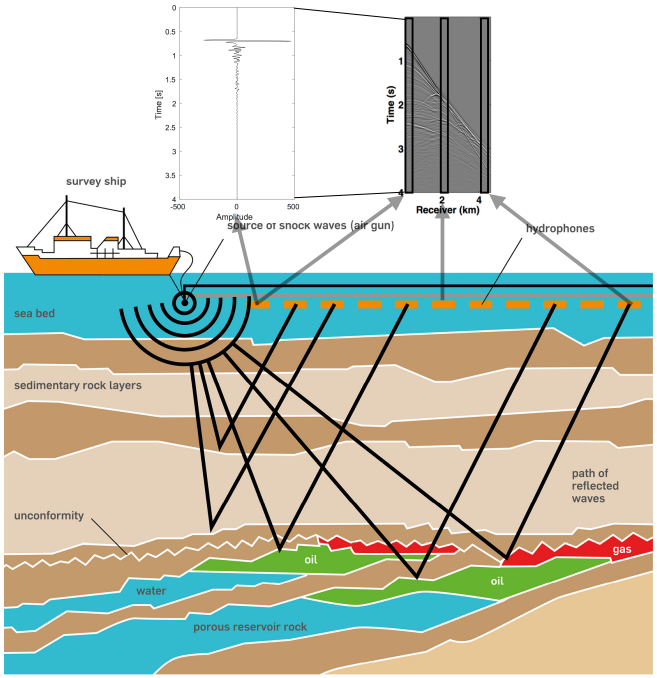


Figure 1.1: Schematic of typical marine seismic acquisition, with towed streamers (cables containing a series of receivers) behind seismic vessels. A array of airguns beneath the water surface produce seismic energy. The offset in seismic experiments is the distance between the source and the receiver, and in 3D seismic experiments the azimuth represents the angle between the source and the receiver in sail-line orientation. At the top of the seismic survey, a seismic trace on the left and one common shot gather on the right are also displayed. Adapted image from KRIENERGY.

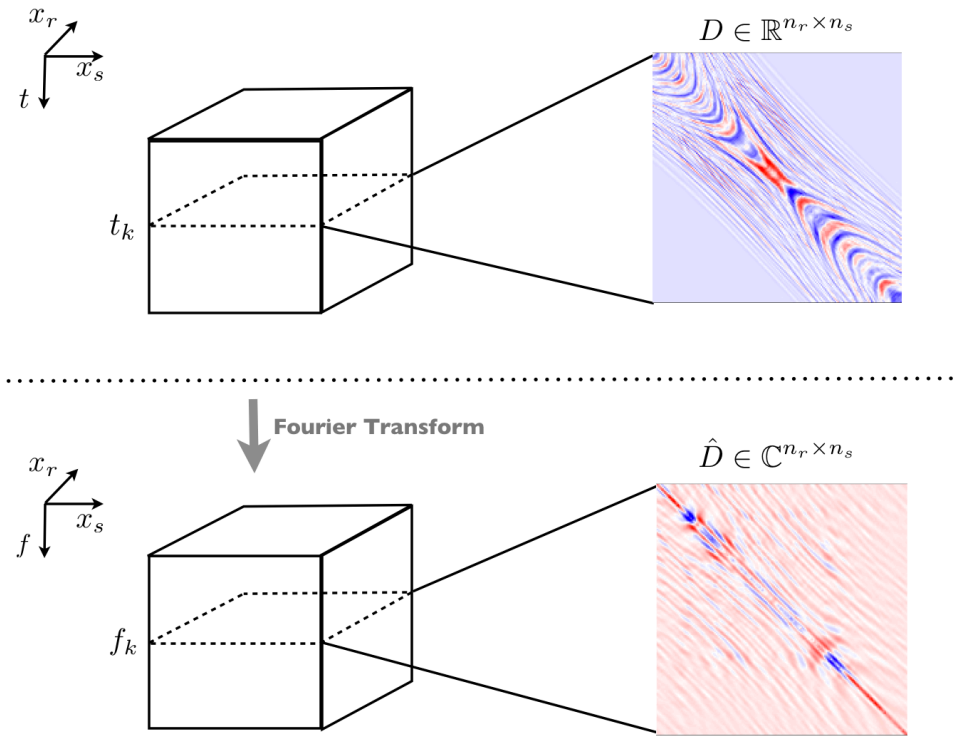


Figure 1.2: Illustration of monochramatic wavefield matrix $\hat{D} \in \mathbb{C}^{n_r \times n_s}$ in 2D seismic survey, where n_r and n_s are the number of receivers and sources repectively. The originally acquired seismic data volume is tranformed from time domain to frequency domain via Fourier tranform. The real component of the data matrix at f_k is displayed.

experiments, this fully sampled acquisition leads to an exponential growth in data size that parallels the size of the interest area. The number of traces in the order of millions for realistically-sized experiments can easily cause terabytes or even petabytes of fully sampled data volumes in size. Aside from the exceedingly time-consuming and cost-intensive process of conducting a 3D seismic survey, full azimuth processing involves massive amounts of data making the storage, management and processing extremely challenging tasks. Recently, low rank techniques have emerged as dimensionality-reduction tools that work by the virtue of an underlying low rank structure of the data. These techniques allow us to facilitate the storage and manipulation on compressed parameters. Low rank methods possess significant advantages over transform-based methods, as they often lead to reduced memory and computational costs because they exploit underlying redundancies that exist in seismic data when organized in a particular way. This redundancy arises from the fact that multiple source experiments ultimately image the same underlying earth.

Unfortunately, terrain limitations or budget constraints often impose restrictions on acquiring fully sampled full-azimuth data in realistic scenarios. To mitigate the issue of coarsely sampled data or missing traces, recent transform-based seismic reconstructions leverage insights from compressive sensing by exploiting transform domain sparsity. The intuitive explanation hinges on seeking the integration of appropriate sampling schemes and sparsifying transforms which behave as random Gaussian matrices (Candès et al. (2006b); Donoho (2006)). Consequently, the data sampling at sub-Nyquist rate under such conditions has high probability to be recovered with sparsity promoting solvers. Hennenfent and Herrmann (2006); Herrmann and Hennenfent (2008) successfully develop curvelet-based recovery algorithms to reconstruct incomplete seismic data volumes with large percentages of missing entries. On the one hand, sampling and storage costs using these methods are no longer dominated by dimensionality of the problems instead scale with transform domain sparsity. On the other hand, if one is interested in query-based access to elements from the data volume, such as common shot/receiver gathers, invariably one has to apply the inverse transform to the vector of transform coefficients before working on it in downstream processings. As a result, reforming the entire data volume in full-azimuth processings prevents practitioners from directly working with seismic data in compressed domain in some sense. Alternative ways to interpolate seismic data are low rank methods, i.e. matrix/tensor completion techniques, which exploit the natural low-rank behavior of seismic data under various permu-

tations (Kumar et al. (2015)). The success of these methods for realistically-sized problems involving fully sampled full-azimuth frequency slices derives from the avoidance of computing prohibitively expensive singular value decompositions (SVDs) on data volumes. Previous work in Kreimer and Sacchi (2012), for instance, represents seismic data in Tucker format and employs a projection on to convex sets (POCS), which suffers from the high computational complexity of SVDs on tensors reshaped as matrices at each iteration. As such, SVD-based projection methods in practice often have to resort to processing on small subsets or windows of the data, which may degrade the reconstructed results (Kumar et al. (2015)). Da Silva and Herrmann (2015) develop an optimization framework to interpolate seismic data in hierarchical Tucker (HT), which eliminates the necessity of computing expensive SVDs by exploiting the smooth manifold structure of this format. As a result, the algorithm boasts the scalability of tensor completion methods to large scale problems.

As summarized above, to fight the *curse of dimensionality* in seismic exploration, low-rank methods can either compress or interpolate seismic data in a storage-efficent scheme under fully sampled or trace missing circumstances, producing greatly-reduced parameters afterwards that might be desirable to ease the burden on storage. Once we represent our industry-scale data in such compressed domain, we in this thesis can design a workflow for directly processing the compressed data in subsequently shot-based processing steps by on-the-fly extraction of common shot/receiver gathers. Using this technique, we will demonstrate how to generate shot/receiver gathers on a per-query basis, i.e. at an arbitrary location as required, without explicitly reforming the data volume. Furthermore, intermediate quantities in the process of shot/receiver extraction can be constructed through computationally efficient matrix-matrix products, and are much smaller than ambient dimensionality. In doing so, we greatly reduce the memory costs involved in storing and processing these data volumes in a subsequent processing context.

1.3 Contribution and Outline

In this work, we demonstrate that possessing a representation of our data volume in a low-rank tensor format will enable us to extract relevant subsets of the data directly from its compressed form. Here we use the Hierarchical Tucker (HT) format (Hackbusch and Kühn (2009); Grasedyck (2010)) to

represent our seismic data, as it requires significantly fewer parameters to represent the data relative to its uncompressed size and one can develop algorithms directly for the compressed parameters using tensor algebra. These benefits outweigh its seemingly complicated construction. We consider two instances of data sampling in this work. First, if our data volume is fully sampled, we use existing, computationally efficient methods to compress a tensor into HT form (Grasedyck (2010)). Second, if our data has missing entries, we use existing methods for interpolating tensors with missing entries in the HT format (Da Silva and Herrmann (2015)). Once our data is in compressed HT form, either through compression or interpolation, we develop an algorithm for extracting arbitrary source or receiver gathers in an on-the-fly manner from the compressed HT parameters, rather than having to form the full data volume explicitly. This approach gives us an oracle for extracting source/receiver gathers at arbitrary locations and reduces the memory costs of working with the full seismic data by two orders of magnitude at the low frequencies. We consider two case studies for integrating our data extraction technique:

- 3D stochastic *full-waveform inversion*
- 3D full subsurface *common-image-point gathers*

This shot extraction procedure only requires efficient matrix-matrix and tensor-matrix products of small parameters matrices and adds little overhead compared to the cost of solving the partial differential equations (PDEs). Our results differ only marginally from using the fully sampled original volume compared to its compressed form while substantially reducing our memory costs. The outline of the methodology proposed in this work is shown in Figure 1.3.

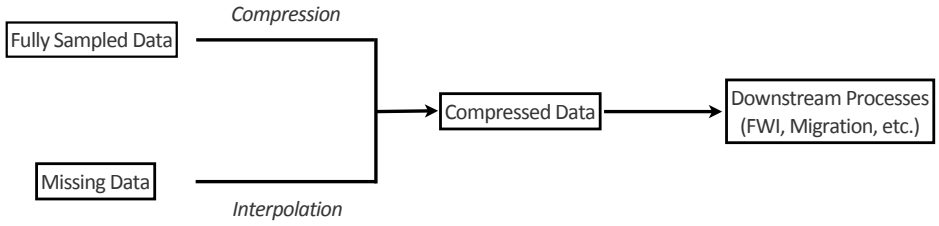


Figure 1.3: Workflow proposed in this thesis.

This thesis is organized as follows:

Chapter 2 describes an extremely memory-efficient HT tensor format applied to seismic 5D data set in a specific permuted domain. Owing to the remarkable compression ratio of this tensor format in seismic low-frequency compennents, we outline a workflow for either compressing or interpolating seismic data sets based on our sampling regimes in compressed HT format. Next, we present the on-the-fly shots/receivers extraction technique directly from significantly-reduced HT parameters, giving query-based access to the full volume without explicitly forming it.

We demonstrate that the combination of massive compression and fast on demand 3D shots or receivers extraction leads to a substantial reduction in memory costs but with minimal effects on results in 3D stochastic full-waveform inversion examples in Chapter 3.

Chapter 4 introduces the probing technique to mitigate the computational bottlenecks in computing the extended image volumes. By using this technique, we show that we can easily apply our memory-efficient extraction approach to form common-image-point gathers without degrading the final results.

Chapter 5 we end up with conclusion of the effective implementation of our proposed framework as well as the potential future works as temporal frequency increases.

Chapter 2

On-the-fly seismic data extraction on a per-query basis

Seismic processing and inversion are challenging problems due to the large amount of data involved and the resulting high computational costs. State-of-art acquisition techniques, such as high density and wide azimuth samplings, are widely used to avoid aliasing and inaccuracy in subsequent processing procedures. Unfortunately, for realistically-sized 3D surveys, the multi-dimensional nature of the data results in the so-called "curse of dimensionality", an exponential explosion in the number of data points as the size of each individual dimension increases. Data sets can easily range from terabytes to petabytes in size and the resulting wave-equation based algorithms must compute solutions to tens of thousands of partial differential equations (PDEs), i.e. the numerical solutions to wave equations. The overwhelming amount of data makes implementing even straightforward, linear-time algorithms daunting for realistic data sets.

2.1 Low-rank techniques in exploration seismology

Low-rank techniques are one approach to mitigate the enormous costs of traditional algorithms working on full data volumes. The underlying idea in this case is that the data volume, when organized as a matrix or tensor in a particular fashion, should have quickly decaying singular values. In this case, the underlying volume can be well approximated by a low-rank matrix or tensor. The number of parameters needed to describe such a low-rank matrix is much smaller than the ambient space, i.e., it is a simpler object

than an arbitrary matrix. Subsequently, operations acting directly on the compressed form of the matrix/tensor are significantly cheaper. Compressed sensing, and likewise matrix completion, theory tells us that we can subsample our data, at a rate commensurate with the underlying dimensionality, i.e., the rank, rather than the ambient space. To recover our original signal, we solve an associated optimization problem, e.g., see (Candès and Recht (2009); Candès and Tao (2010); Recht (2011)).

There has been a surge of interest in recent years in applying low-rank techniques to seismic data problems, including interpolation (Ma (2013); Kumar et al. (2015); Aravkin et al. (2014); Trickett et al. (2010)), noise attenuation (Freire and Ulrych (1988); Bekara and Van der Baan (2007); Nazari Siahsar et al. (2016)), estimation of primaries by sparse inversion (Jumah and Herrmann (2014)), simultaneous source deblending (Cheng and Sacchi (2015); Kumar et al. (2016)), and travel-time tomography (Stork (1992)). Extensions of these low-rank ideas to multi-dimensional tensors in the seismic context can be found in, e.g., (Kreimer and Sacchi (2012); Kreimer et al. (2013); Trickett et al. (2013); Da Silva and Herrmann (2015)).

Low-rank methods are not the only technique to represent high dimensional wavefields in a low-dimensional manner. Transform-based methods consider a representation of the data volume in a domain such as wavelets (Villasenor et al. (1996)) or curvelets (Herrmann et al. (2007); Herrmann and Hennenfent (2008)) and store only a small subset of the total coefficients. While sufficient for the purposes of storing, retrieving, or interpolating data, these methods become particularly cumbersome if one is interested in extracting specific subsets from the compressed volume. Furthermore, for curvelets, since they are four times redundant in 2D (Candès et al. (2006a)) and five times redundant in 3D (Ying et al. (2005)), the benefits of sparsity are somewhat dwarfed by the need to handle such large coefficient vectors in memory, while other orthogonal transforms, such as wavelets, typically compress seismic waveforms much less efficiently than curvelets in a relative sense. Transform-based methods can also be much more complicated to implement compared to low-rank methods, the latter of which merely involves efficient matrix-matrix multiplications.

In practice, acquired data frequently contains missing entries, either due to budget, time, or environmental constraints. There are a variety of mathematical techniques used to estimate the fully sampled data volume in this case. By exploiting the sparsity or correlations among coefficients, transform-based approaches can be used for interpolation, in the

case of representations such as Radon (Kabir and Verschuur; Wang et al. (2010)), Fourier (Sacchi et al. (2009); Curry (2010)), wavelets (Villasenor et al. (1996)), and curvelets (Hennenfent and Herrmann (2006); Herrmann and Hennenfent (2008)). These methods have been successful for interpolation, but the computational costs of these approaches can be large. Recent development in matrix completion (Oropeza and Sacchi (2011); Kumar et al. (2015)) and tensor completion (Kreimer and Sacchi (2012); Trickett et al. (2013); Da Silva and Herrmann (2015)) techniques can substantially lower these costs. One must take care when applying these methods naively, however, as using an off-the-shelf singular value decomposition (SVD)-based implementations is extremely computationally expensive, even in domains that promote singular value decay such as midpoint-offset. As a result, practitioners using SVD-based methods often resort to working with small subsets of the data at a time, i.e., windowing. As demonstrated in Kumar et al. (2015), the act of windowing can even significantly degrade the quality of the recovered data.

2.2 Hierachical Tucker representation

Multilinear or tensor algebra is a branch of computational mathematics that has become increasingly ubiquitous in the era of "big data". As noted previously, the number of entries of a tensor grows exponentially with increasing number of dimensions, i.e., so-called "curse of dimensionality", which incurs substantial memory and computational cost to process such objects. As such, various low-rank tensor formats, such as Canonical Polyadic (CP) format (Carroll and Chang (1970)) and the Tucker format (De Lathauwer et al. (2000)), have been developed to exploit redundancies among the various dimensions and represent the full tensor in a more compact manner. The Hierarchical Tucker tensor format is a novel structured tensor format introduced in (Hackbusch and Kühn (2009)), which results in the number of HT parameters growing linearly with the number of dimensions rather than exponentially, which is extremely storage-efficient and computationally tractable for parametrizing high-dimensional problems.

2.2.1 Notation and Preliminaries

We represent vectors (i.e., one-dimensional quantities) as boldfaced lowercase letters, e.g., $\mathbf{x}, \mathbf{y}, \mathbf{z}$ and matrices and tensors (i.e., multi-dimensional

quantities) as boldfaced uppercase letters, e.g., $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. We use $\mathbf{X}^*$ to denote the conjugate transpose of $\mathbf{X}$ and $\mathbf{X}^T$ to denote the real transpose of $\mathbf{X}$, and $\|\cdot\|$ stands for ℓ_2 norm. The vectorization operator vec stacks the columns of a matrix, or the multiple dimensions of a tensor, into a vector. $\mathbf{A} \otimes \mathbf{B}$ denotes the Kronecker product between matrices $\mathbf{A}$ and $\mathbf{B}$. I_n denotes the identity operator of size n. The complex conjugate of a vector $\mathbf{z}$ is written as $\bar{\mathbf{z}}$.

The *matricization* of a $d-$dimensional tensor $\mathbf{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$, along the $i-$th coordinates, denoted $\mathbf{X}^{(i)}$, reshapes the tensor so that the coordinates corresponding to the indices i are along the rows and the remaining indices are along the columns. For example, for a 5D tensor $\mathbf{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3 \times n_4 \times n_5}$, $\mathbf{X}^{(1)}$ is a matrix with the size $n_1 \times n_2 n_3 n_4 n_5$ the first dimension along the rows and dimensions 2,3,4,5 along the columns. Similarly, $\mathbf{X}^{(1,3)}$ is an $n_1 n_3 \times n_2 n_4 n_5$ matrix has the two dimensions 1,3 along the rows and dimensions 2,4,5 along the columns.

Given a $d-$dimensional tensor $\mathbf{X} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ and a linear operator $\mathbf{A} \in \mathbb{C}^{m_i \times n_i}$, the *multilinear product* of $\mathbf{X}$ and $\mathbf{A}$ in the $i-$th dimension, denoted $\mathbf{Y} = \mathbf{A} \times_i \mathbf{X}$, is defined in terms of the matricizations

$$\mathbf{Y}^{(i)} = \mathbf{A}\mathbf{X}^{(i)} \tag{2.1}$$

or in terms of Kronecker products as

$$\text{vec}(\mathbf{Y}) = (I_{n_d} \otimes I_{n_{d-1}} \otimes \cdots \otimes I_{n_{i+1}} \otimes A \otimes I_{n_{i-1}} \otimes \cdots \otimes I_1)\text{vec}(\mathbf{X}). \tag{2.2}$$

In words, we are matricizing the tensor along the $i-$th dimension, applying the operator $\mathbf{A}$, and then reshaping this matrix into a tensor. In seismic terms, this concept encapsulates many familiar operations. For instance, if one has a data volume $\mathbf{D}$ with coordinates (source x, source y, receiver x, receiver y, time), performing a Fourier transform along the time axis, with Fourier matrix $\mathbf{F}_t$, would be written as $\mathbf{F}_t \times_5 \mathbf{D}$.

2.2.2 Dimension tree and storage analysis

Although it is slightly technical to define the HT format explicitly, we define some preliminary components below. Given a $d-$dimensional tensor, we associate a *dimension tree* $\mathcal{T}$ as a binary tree with the root being assigned the labels $t_{\text{root}} = \{1, 2, \ldots, d\}$. Every node t is assigned a set of labels $t \subset \{1, \ldots, d\}$ and its left and right children, denoted t_l and t_r, respectively,

form a disjoint partition of t, i.e., $t = t_l \cup t_r$, $t_l \cap t_r = \emptyset$. The dimension tree specifies how we *separate* groups of dimensions from each other, where "separate" is understood in the sense of the SVD. An example of a dimension tree for a 6−dimensional tensor is shown in Figure 2.1. Here we interpret this figure as dimensions $\{1, 2, 3, 4\}$ being separated from dimensions $\{5, 6\}$. Dimensions $\{1, 2\}$ of the tensor are further split apart from dimensions $\{3, 4\}$ on the left side of the tree, and so on. We shall see a specific instance of a dimension tree for seismic data later in this work.

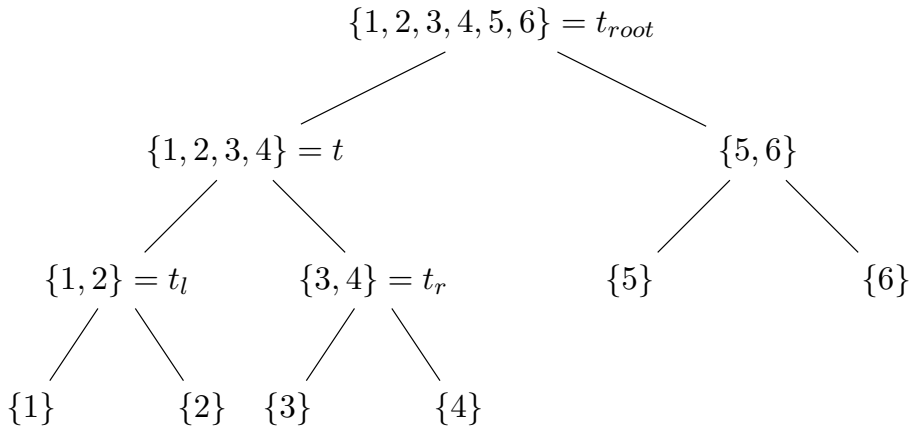


Figure 2.1: A dimension tree for $\{1, 2, 3, 4, 5, 6\}$, from (Da Silva and Herrmann, 2015).

In lieu of an elaborate technical definition of the HT format, which can be found in (Grasedyck (2010)), we instead present a pictorial representation of its construction in Figure 2.2. For a 4−dimensional tensor $\mathbf{X}$, we consider *matricizing* it along its first two dimensions, resulting in $\mathbf{X}^{(1,2)}$. We can consider an "SVD-like" decomposition of this matrix, which splits it in to a product of three smaller matrices. One can understand the "SVD-like" decomposition from the perspective of matrix decompositions for general HT format, which does not necessarily require an orthonormal basis. Noticing that the matrix $\mathbf{U}_{(1,2)}$ contains dimensions $(1, 2)$ of the tensor along the rows, we can further reshape it in to a $n_1 \times n_2 \times k_{1,2}$ cube that can be further decomposed in this multilinear fashion. We apply the same recursive splitting to the matrix $\mathbf{U}_{(3,4)}$, although it is not shown. As a result of this recursive construction, the intermediate matrices $\mathbf{U}_{(1,2)}, \mathbf{U}_{(3,4)}$, which contain multiple spatial dimensions and hence are onerous to construct, do

not have to be formed explicitly. Instead, once we have knowledge of the small matrices $\mathbf{U}_i$, $i = 1, 2, 3, 4$ and small 3−dimensional tensors $\mathbf{B}_i$, $i = (1, 2), (3, 4), (1, 2, 3, 4)$, running this multilinear construction in reverse will reconstruct the entire tensor $\mathbf{X}$. As a result, the total number of parameters needed to specify a d−dimensional HT tensor is bounded above by $dNK + (d-2)K^3 + K^2$, where N is the maximum spatial sampling in all dimensions and K is the maximum rank. Tensors that are able to be represented in the HT format have significantly fewer parameters than the N^d parameters required to represent the full data. For instance, if $N = 100, d = 4, K = 20$, then the number of HT parameters needed to represent the tensor is 24400, compared to 10^8 for the pointwise array data.

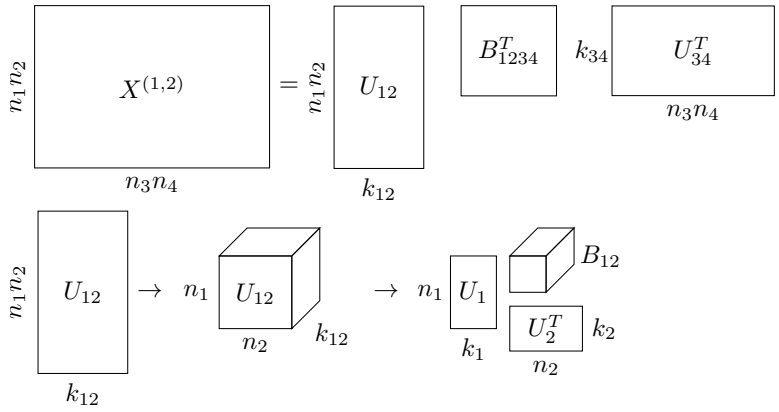


Figure 2.2: Hierarchical Tucker format for a 4D tensor $\mathbf{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3 \times n_4}$., image from (Da Silva and Herrmann, 2013).

2.3 HT format for seismic 5D data set

In the seismic context, our data has five dimensions (source x, source y, receiver x, receiver y, time). We process individual temporal frequency slices, one at a time, resulting in data volumes under consideration with dimensions (source x, source y, receiver x, receiver y). For the purpose of compression, we need to ensure that the data volume has quickly decaying singular values in each of the relevant matricizations. Noting that the organization of the tensor has a major impact on its low-rank nature, we typically permute our data from canonical organization (source x, source y, receiver x, receiver y) into a non-canonical organization (source x, receiver x, source y, receiver

y), which leads to faster decaying singular values for the associated matricizations (Aravkin et al. (2014); Da Silva and Herrmann (2015)). This permutation results in a data volume that is much more amenable to compression in the HT format compared to the standard (source x, source y, receiver x, receiver y) organization. One potential reasoning is contained in Demanet (2006) , where the author uses this organization in the context of compressing solution operators of the wave equation. Since the data itself is the Green's function of the wave equation restricted to the surface, we find that this non-canonical organization enables fast singular value decay of the data volume. Moreover, from the perspective of low-rank recovery, considering randomly missing sources or receivers in the non-canonical ordering results in growth of the singular values in the corresponding matricizations of the tensor, leading to more favourable reconstruction conditions, as in Kumar et al. (2015).

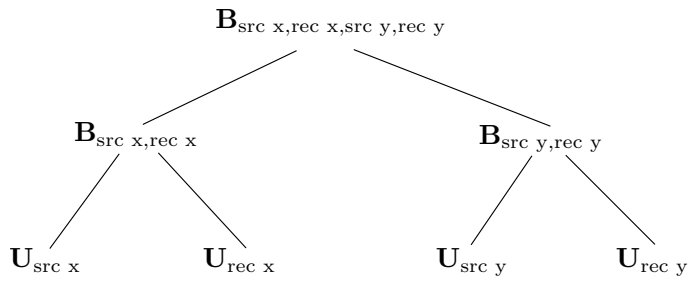


Figure 2.3: Non-canonical dimension tree for the HT format applied to each seismic 4D monochromatic slice.

We visualize the mathematical decomposition depicted in Figure 2.2 using actual seismic data, shown in Figure 2.4, Figure 2.5 and Figure 2.6. For brevity, we abbreviate "source" as "src" and "receiver" as "rec" in this image. As indicated previously, we first reshape our seismic data in the non-canonical matrix ordering, and subsequently can decompose this matrix into a product of three matrices. Note that, although the SVD is used to decompose this tensor, these resulting matrices are not necessarily constrained to have the same orthogonality/diagonal structure as the SVD matrices in the

general HT format. From the matrix $U_{\text{src x, rec x}}$, we can consider each column as a vectorized matrix. Reshaping this vector into a matrix allows us to further decompose it in an SVD-like manner. The low number of parameters of this format arises in part by keeping the $U_{\text{src x}}$ and $U_{\text{rec x}}$ matrices constant for decomposing each column of $U_{\text{src x, rec x}}$. It is not straightforward to derive physical insight from these quasi-singular vectors, although they do tend to qualitatively behave as Green's functions. For a more thorough treatment of the HT format, as well as attendant software, we refer the interested reader to Da Silva and Herrmann (2015).

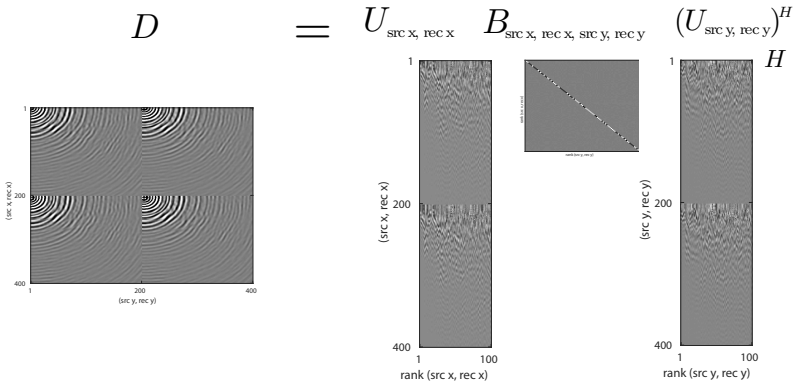


Figure 2.4: Visualizing the HT format on seismic data. Axes zoomed for visibility.

2.3.1 Seismic data compression

When our data are fully sampled, we can use the method of Tobler (2012) to truncate the full volume to the HT tensor format. This method allows us to prescribe an error level and a maximum rank parameter to approximate the data. We outline the performance of this algorithm in Table (2.1) on a synthetically generated data set on the 3D Overthrust model with 50 x 50 sources and 396 x 396 receivers. As there is no unique notion of (minimal) rank for tensors, the truncation algorithm merely upper bounds the error by choosing appropriate intermediate ranks automatically. As the temporal

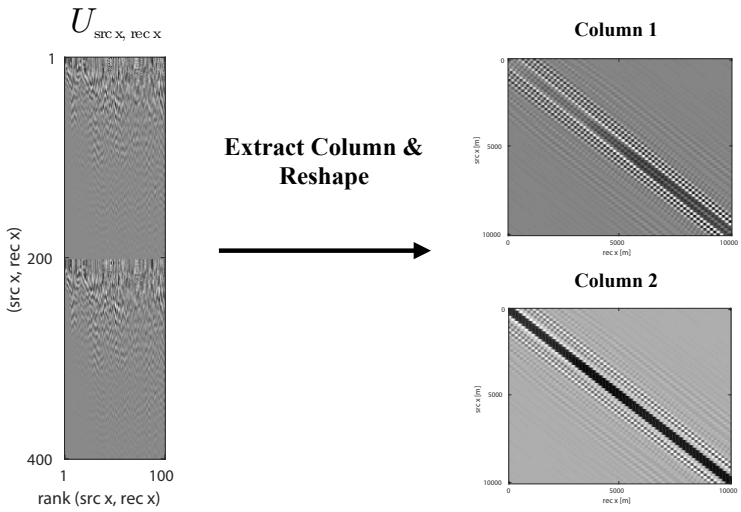


Figure 2.5: Visualizing columns of the intermediate matrices of the HT format. Each column can be reshaped into a matrix with dimensions $n_{\text{rec x}} \times n_{\text{src x}}$. Axes zoomed for visibility.

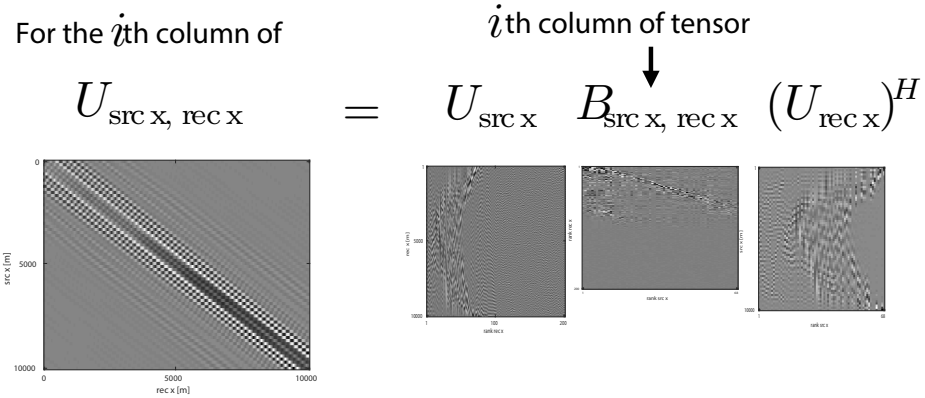


Figure 2.6: Visualizing the decomposition of the columns from Figure 2.5. The leaf matrices $U_{\text{src x}}, U_{\text{rec x}}$ remain constant across the columns of $U_{\text{src x, rec x}}$ while the intermediate matrix $B_{\text{src x, rec x}}$ is allowed to vary for each column. Axes zoomed for visibility.

	Frequency (Hz)	Parameter size	Compression Ratio	SNR
Non-canonical	3	71 MB	98.8%	42.8 dB
Canonical	3	501 MB	91.6%	42.9 dB
Non-canonical	6	421 MB	92.9%	43.0 dB
Canonical	6	1194 MB	79.9%	43.1 dB

Table 2.1: Compression rates of the HT truncation method at different frequencies, in different data organizations. The original frequency slice is approximately 5.8GB in size. SNR $= 20\log_{10}(\frac{\|D-\tilde{D}\|_2}{\|D\|_2})$.

frequency increases, so do the internal ranks of the tensor format, and thus lower frequency data benefits from compression less than higher frequency data. As noted previously, the canonical organization of the data performs much more poorly than the non-canonical permutation and the difference becomes more apparent at higher frequencies. We focus on using the non-canonical organization of the data for the remainder of this work.

2.3.2 Seismic data interpolation

As discussed above, terrain restrictions or cost limitations almost always limit fully sampled data in realistic scenarios. In order to compensate for this missing data, we apply the algorithm described in Da Silva and Herrmann (2015) to reconstruct the full data volume by solving the optimization problem

$$\min_x \|\mathcal{A}\phi(x) - b\|^2. \tag{2.3}$$

where x is the vectorized set of HT parameters $(\mathbf{U}_t, \mathbf{B}_t)$ from previously, ϕ maps x to the fully-expanded tensor $\phi(x)$, as in the reverse process of Figure 2.2, $\mathcal{A}$ is the subsampling operator, and b is our subsampled data. This algorithm can interpolate each 4D monochromatic frequency slices quickly, as it does not compute SVDs on large matrices, and it can successfully recover seismic data volumes from a high level of randomly missing data. When our data has randomly missing entries, we use these efficient algorithms to recover an estimate of the fully sampled data in compressed HT form. We refer the reader who is interested in the details of these algorithms to Da Silva and Herrmann (2015).

2.4 On-the-fly extraction of shot/receiver gathers

Irrespective of our sampling regime, once we have HT representation of our data volume, we can greatly reduce the computational and memory costs of working with our data. To make full use of the data directly in its compressed form, we present an approach to extract a shot (or receiver) gather at a given source location (i_x, i_y) directly from the compressed parameters. Here, we use Matlab colon notation $\mathbf{A}(i,:)$ to denote the extraction of the i^{th} row of the matrix $\mathbf{A}$, and similarly for column extraction. The common shot gather can be extracted by computing

$$\begin{aligned}
\mathbf{U}_{\text{sx,rx}} &= \mathbf{U}_{\text{sx}}(i_x,:) \times_1 \mathbf{U}_{\text{rx}} \times_2 \mathbf{B}_{\text{sx,rx}} \\
\mathbf{U}_{\text{sy,ry}} &= \mathbf{U}_{\text{sy}}(i_y,:) \times_1 \mathbf{U}_{\text{ry}} \times_2 \mathbf{B}_{\text{sy,ry}} \\
\mathbf{D}_{i_x,i_y} &= \mathbf{U}_{\text{sx,rx}} \times_1 \mathbf{U}_{\text{sy,ry}} \times_2 \mathbf{B}_{\text{sx,rx,sy,ry}}
\end{aligned} \tag{2.4}$$

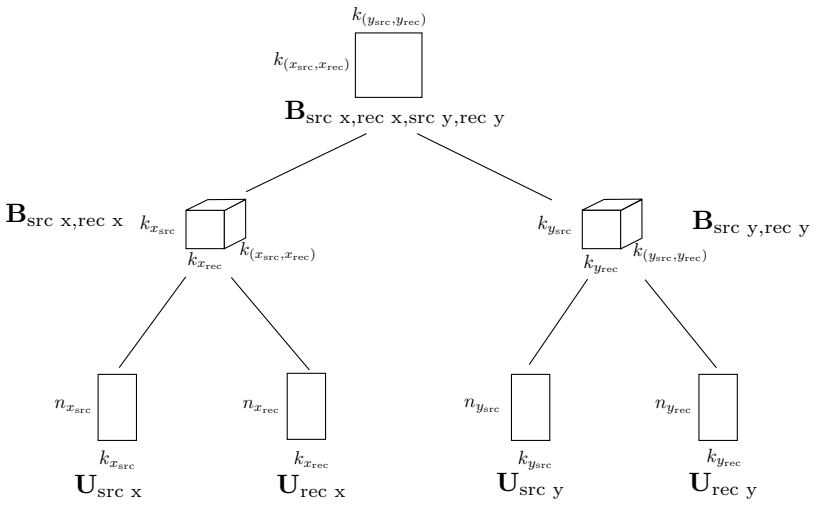
This algorithm follows the main construction of the parameters to full tensor mapping (Grasedyck (2010)), although specified to a single shot location. A pictorial representation of this algorithm is given in Figure 2.7 - Figure 2.9. Most importantly, at no point do we need to form any intermediate quantities of the size of the full tensor and all the computations in equation (2.4) can be implemented via multilinear products, outlined in Algorithm 2.1. This allows us to efficiently have query-based access to the data volume in its compressed form, which will be useful for the stochastic FWI approach detailed below. Note that common receiver gather extraction can be implemented in an analogous way.

Algorithm 2.1 Extracting a common shot gather from compressed HT parameters

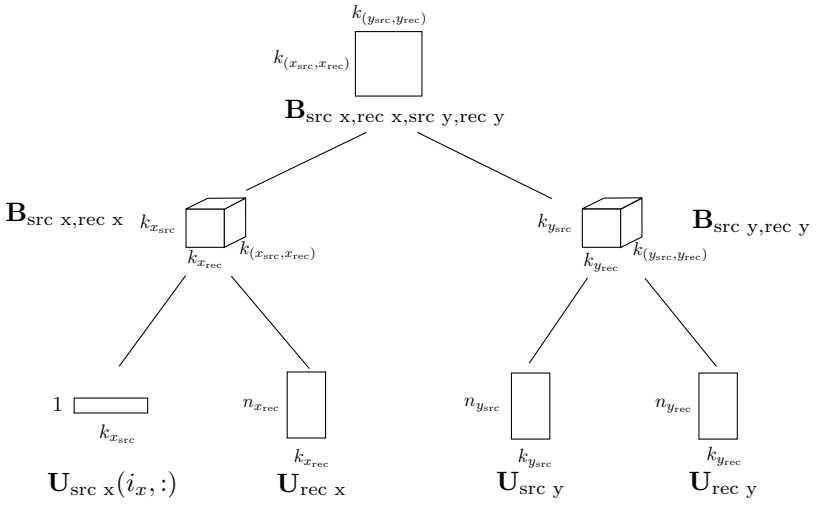
Input: Indices of the source coordinates i_x and i_y, and dimension tree

1. Extract the vector $\mathbf{u}_{srcx}$ from the matrix $\mathbf{U}_{srcx}(i_x,:)$
2. Multiply $\mathbf{B}_{srcx,recx}$ along the k_{srcx} dimension with the vector $\mathbf{u}_{srcx}$ (in the sense of multilinear product)
3. Multiply the matrix obtained from step 2 along the k_{recx} dimension (second dimension) with $\mathbf{U}_{recx}$, resulting in a matrix $\mathbf{U}_{srcx,recx}$ of size $n_{recx} \times k_{(srcx,recx)}$
4. Repeat steps $1, 2, 3$ along the y coordinate to obtain the matrix $\mathbf{U}_{srcy,recy}$ of size $n_{recy} \times k_{(srcy,recy)}$
5. The product $\mathbf{U}_{srcx,recx} B_{srcx,recx} U^H_{srcy,recy}$ results in the final shot gather

Figure 2.10 and Figure 2.11 demonstrate the successful interpolation of the data volume from a high level of missing data, resulting in shot-gathers that are simple to extract with Algorithm 2.1.

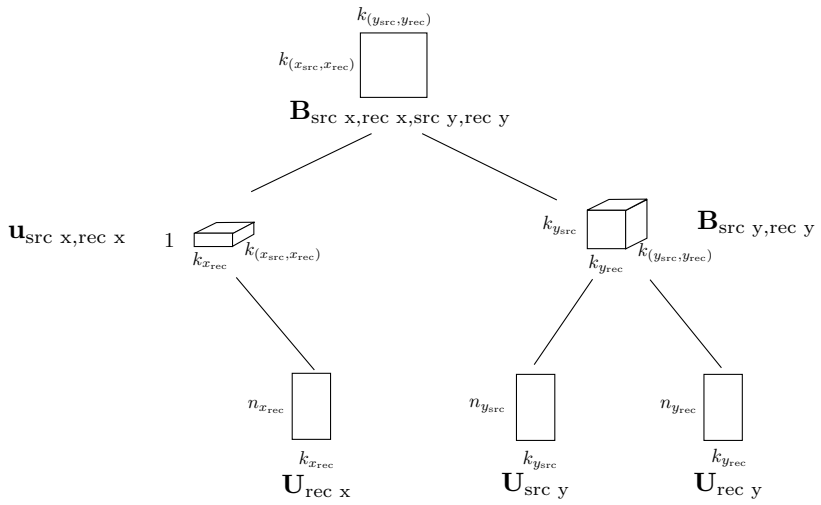


(a)

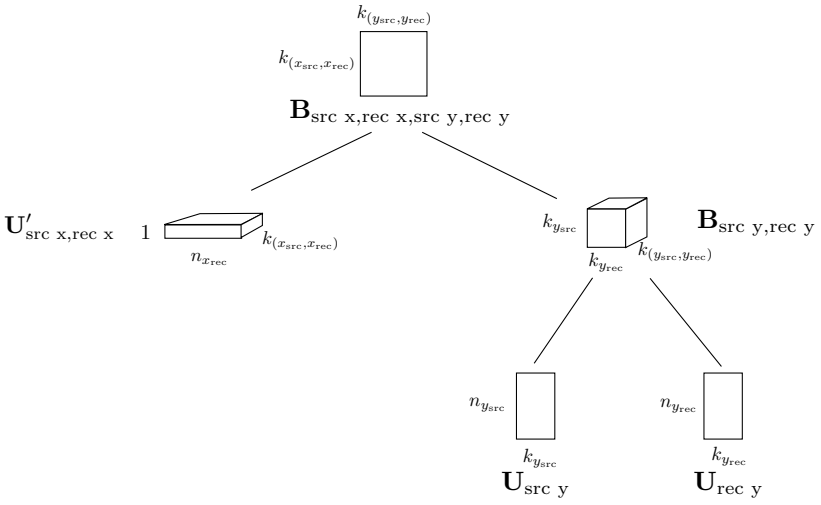


(b)

Figure 2.7: A pictorial depiction shot extraction of Algorithm 2.1. (a) Initial input parameters, and (b) Step 1

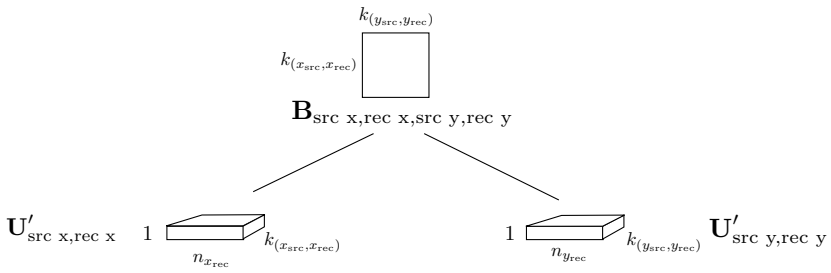


(a)

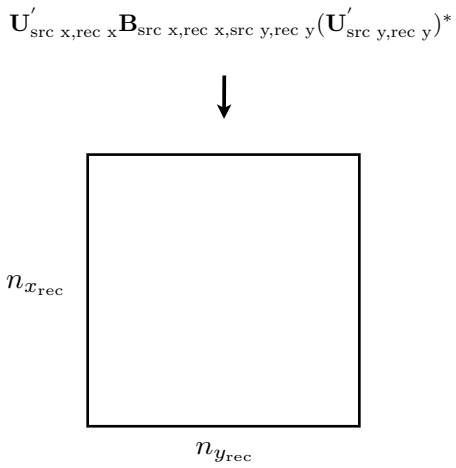


(b)

Figure 2.8: A pictorial depiction shot extraction of Algorithm 2.1. (a) Step 2, and (b) Step 3



(a)



(b)

Figure 2.9: A pictorial depiction shot extraction of Algorithm 2.1. (a) Step 4, and (b) A common shot gather extracted at a specified location.

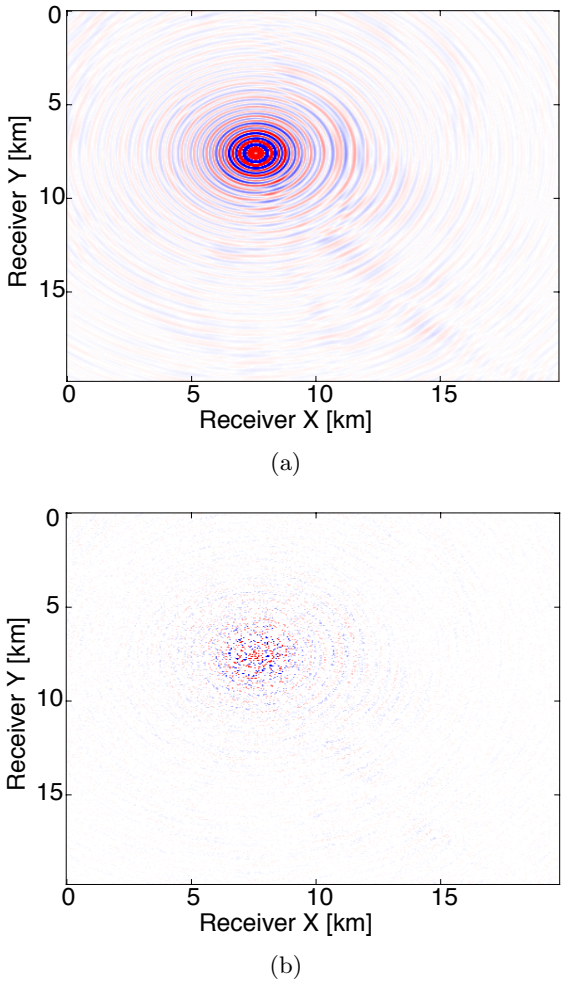
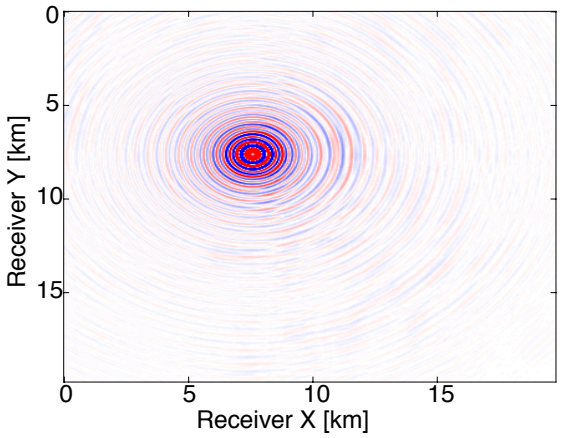
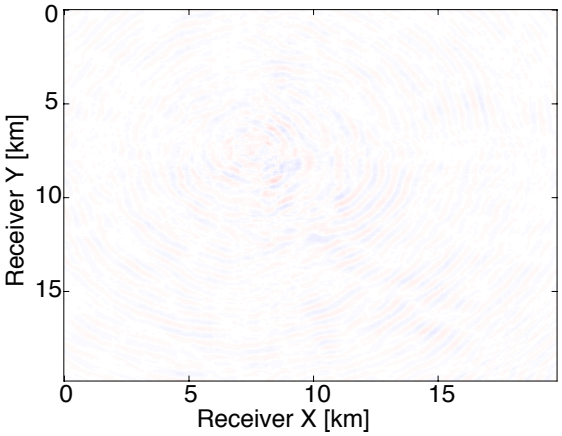


Figure 2.10: Extracted shot from the Overthrust data at 6Hz, full data vs compressed data with Algorithm 2.1 after HT interpolation. (a) True data, and (b) Missing 80% random entries.



(a)



(b)

Figure 2.11: Extracted shot from the Overthrust data at 6Hz, full data vs. compressed data with Algorithm 2.1 after HT interpolation. (a) Extracted from compressed, recovered data, and (b) Difference between the true and reconstructed data.

Chapter 3

Implementation of inversion with Hierachical Tucker

Migration or seismic imaging, a process which repositions measured seismic data on the surface into their "true" subsurface locations under investigation, is a most powerful tool to help engineers for illumination of complicated geological structures. However, the quality of final imaging results are critically dependent on the accurate estimation of velocity model since inaccurate velocity input leads to reflectors in wrong positions through the migration. Given the measured seismic data on the surface, the *inversion* is to find the medium parameters such as velocity that can reasonably explain the physical mechanisms. As such, this process is regarded as the reverse of the *forword modelling* in which our solution is a synthetic wavefield (or seismic section) based on the provided medium paremeters.

Traveltime tomography (Bording et al., 1987; Taillandier et al., 2009), using arrival time information, is one of the most widely implemented techniques to invert medium parameters (slowness or velocity) in industries. The arrival time information can be extracted from seismic data either manually or antomatically (Saragiotis et al., 2013). Traveltime tomography, based on the eikonal equation or ray-tracing techniques, falls into the class of high-frequency approximation solution of the wavefield. As a result, such kinds of ray-based tomography methods are valid when velocity scale length in the medium is much larger than the seismic wavelength. Contrarily, the dominant behavior becomes diffraction in anomalous velocity region, and the propagating wavefront of the seismic waves therefore can not be correctly described by ray paths (Schuster, 2007).

Driven by its superb ability to produce high resolution of the velocity model building, another data domain method called *full waveform inversion* (Tarantola, 1987; Pratt, 1999; Krebs et al., 2009; Li et al., 2012; Biondi and Almomin, 2014) has increasingly gained broader popularity in both

industries and academia in recent years. In this chapter I cover the basic idea of full waveform inversion, and I will then describe the reduction in computational and memory costs by the combination of a novel optimization technique with our developed query-based extraction approach.

3.1 Full waveform inversion

Full waveform inversion (FWI) is a non-linear data-fitting procedure that estimates a model of the subsurface given measurements made on the Earth's surface. Mathematically, we find a model $\mathbf{m}$ that generates predicted data that best agrees with our observed data in a least-squares sense, i.e.,

$$\min_{\mathbf{m}} \Phi(\mathbf{m}) = \frac{1}{2N_s} \sum_{i=1}^{N_s} \|\mathbf{P}_r \mathbf{H}(\mathbf{m})^{-1} \mathbf{q}_i - \mathbf{d}_i\|^2, \tag{3.1}$$

where $\mathbf{P}_r$ maps the computed wavefield from the subsurface to the receiver locations, $\mathbf{u}_i = \mathbf{H}(m)^{-1}\mathbf{q}_i$ is the wavefield corresponding to the $i-$th source $\mathbf{q}_i$, and $\mathbf{d}_i$ is the observed data, and we have N_s sources. Our model of the wave equation in this case is the constant-density acoustic Helmholtz equation, although other models are possible. We consider inverting a single frequency at a time, without loss of generality. This optimization problem in Equation (3.1) is typically solved by using iterative algorithms with the model update

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \mathbf{s}_k, \tag{3.2}$$

where $\mathbf{s}_k$ denotes the search direction and α_k the step size at each iteration. The simplest case is the steepest-descent method in which the search direction is chosen as the negative gradient of the objective function, i.e. $\mathbf{s}_k = -\nabla\Phi(\mathbf{m})$. Computing the full gradient $\nabla\Phi(\mathbf{m})$ is computationally daunting when N_s is large as we have to solve N_s partial differential equations (PDEs) at each iteration. For the wave-equation based inversion algorithms, these PDEs in the frequency domain are typically linked with solving numerical Helmholtz equations.

3.2 Stochastic optimization

We follow the algorithmic developments in Da Silva and Herrmann (2016) and use a parallel stochastic optimization approach to improve convergence

with limited passes through the data. If our computational environment has p independent parallel processes with $p \ll N_s$, we partition the data in to p disjoint subsets of size $N_s/p > 1$. Each node j has access to its own subset of data indexed by $I_j \subset \{1, \ldots, N_s\}$. At every outer iteration of our full waveform inversion algorithm, we choose a random subset $\tilde{I}_j \subset I_j$ at each node, where $|\tilde{I}_j| < |I_j|$ (in our experiments, we choose $|\tilde{I}_j| = 1$, so we choose a single shot per node). We then use an L-BFGS method (Schmidt et al. (2009)) to approximately solve the following subproblem

$$\min_m \tilde{\Phi}(\mathbf{m}) := \frac{1}{2p} \sum_{j=1}^{p} \frac{1}{|\tilde{I}_j|} \sum_{k \in \tilde{I}_j} \|\mathbf{P}_r \mathbf{H}(\mathbf{m})^{-1} \mathbf{q}_k - \mathbf{d}_k\|^2. \tag{3.3}$$

We repeat this procedure for T outer iterations, redrawing the shots we use in the subproblem at each iteration. Applying the L-BFGS method directly in a stochastic context results in computing differences of gradients that correspond to different subsets of shots. Computing this difference is not representative of the full, non-stochastic Hessian, upon which the LBFGS method relies, and therefore results in poor convergence. This approach, on the other hand, allows us to avoid this difficulty, as we use the LBFGS method directly to solve the subproblem and postpone our redrawing of shots until the algorithm moves on to another subproblem. Other approaches to stochastic LBFGS (Moritz et al., 2016; Gower et al., 2016) have nice theoretical properties but may be computationally expensive for FWI. We limit the number of L-BFGS iterations for the subproblem so that the number of PDEs used to solve the subproblem is equivalent to computing the objective + gradient with all of the data (i.e., each subproblem is equivalent in cost to one full pass through the data). In this manner, we can drastically reduce the per-iteration cost of FWI while still respecting utilizing our parallel resources effectively. A standard stochastic gradient algorithm, by comparison, would select a random subset of all shots to process at a given time, which would not respect the distribution of data among the workers and therefore require either costly data redistribution between nodes or result in some nodes remaining idle while others are working. In the case of full-azimuth data acquisition, we can also use a modification of the previous technique to extract simultaneous sources from the compressed HT parameters. In its uncompressed format, one would have to sum over all shots, distributed over a number of nodes, with random weights, which would incur a large communication overhead. Our compressed approach does not incur this overhead, which is important when scaling these algorithms to large parallel environments.

3.3 Numerical examples on Overthrust model

To demonstrate that our method significantly reduces memory costs of working with the data volume, our first trial of full-waveform inversion is on the SEG/EAGE 3D Overthrust velocity model (Figure 3.1). This $20km \times 20km \times 5km$ velocity model, discretized on a $50m$ grid in each direction, contains structurally complex areas such as fault belts and channels. We modify this model by adding a $500m$ water layer on top. We generate our data from 4 frequencies between $3Hz$ and $6Hz$ with spacings of $1Hz$ using a 50×50 source grid with $400m$ spacing and a 396×396 receiver grid with $50m$ spacing on the ocean bottom. The size of each frequency slice is roughly $5.8GB$. From the full data, we randomly remove 80% of receivers from each frequency slice, which forms the subsampled data at each of these frequencies we are working on. We therefore for the remainder of this work use the term "subsampled data" specially to refer to the fully sampled and full azimuthal seismic data of removing a large percentage of random receivers. We then are able to obtain our compressed data volume for each monochromatic slice by interpolating in the HT format.

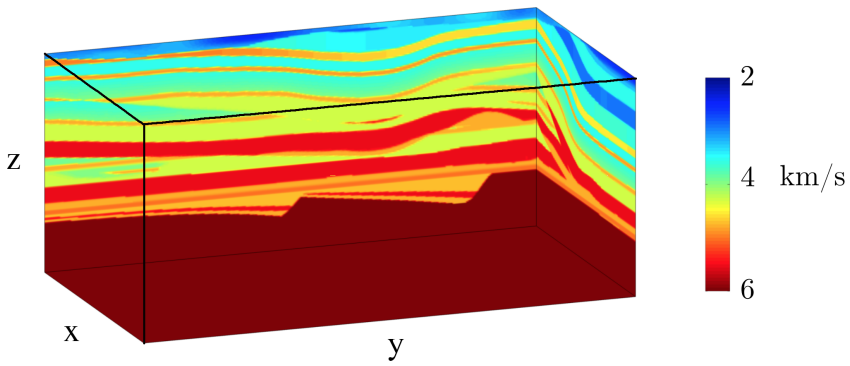


Figure 3.1: A part of SEG/EAGE 3D Overthrust velocity model.

In this case, we invert a single frequency at one time using 50 nodes with 8 Matlab workers running for each, where each node has 20 CPU cores and $256GB$ of RAM. We run the 3D FWI experiments for both the full data and compressed HT data recovered from interpolation, fixing the total number of PDEs solved. Figure 3.2 - Figure 3.5 show inversion results for both full data and compressed data. Despite working with heavily subsampled initial

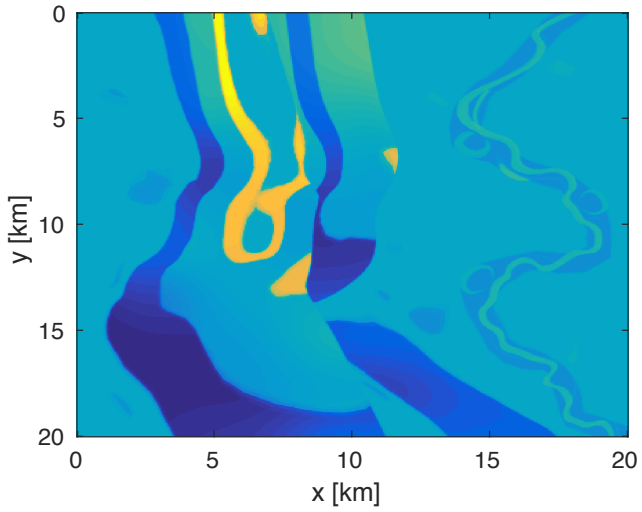
data, the interpolation algorithm is able to accurately estimate the entire data volume and the compressed parameters are used to successfully invert the velocity model at a greatly reduced memory cost. In this case, the data volume sizes are reduced by over 90%, with little visual difference in the final inversion results.

3.4 Numerical examples on BG model

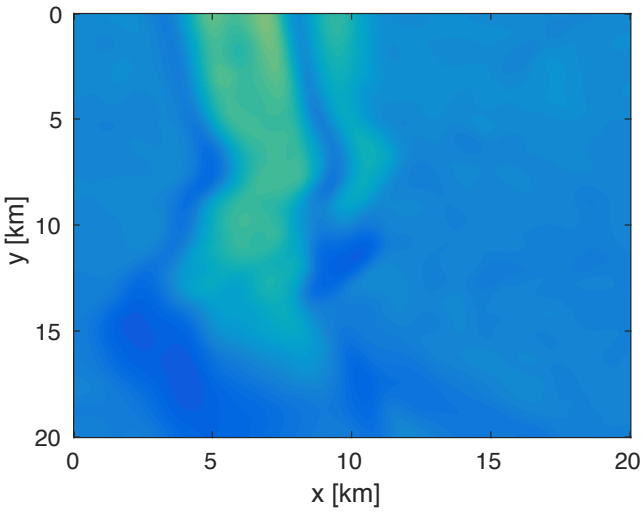
To further consider the velocity model with more complicated sedimentary variability and also cover two sampling scenarios in our designed workflow, we integrate our data-extraction approach with the aforementioned stochastic FWI method and perform inversion on a truncated portion of the BG Compass model. We select a 10 km x 10 km x 1.8 km central portion of the model which has 50m x 50m x 12m grid spacing. We place a 49 x 49 grid of sources along the top of the model and a 196 x 196 grid of receivers in an ocean-bottom node setup, using a Ricker wavelet with peak frequency at 10Hz. We use frequency-domain inversion code from (Da Silva and Herrmann, 2016), inverting a single frequency at a time, from 3Hz to 6Hz in 0.25Hz increments. As our maximum inverted frequency is quite low, we do not expect to recover a highly detailed model, but instead focus on the differences between the inversion with the original data and its compressed version. We limit the number of passes through the full data to $T = 3$. Despite these limited number of passes through the data, our algorithm makes significant progress towards the true solution. Various 2D slices through the true and initial models are shown in Figure 3.6, 3.8, and 3.10.

3.4.1 Full data vs. Compressed data

In the "full data" scenario, where our data has full source and receiver coverage, we compare the inversion results using the full data as well as the HT compressed data, which are shown in Figure 3.7, 3.7, and 3.11. Despite our data being compressed by over 90%, the inversion of compressed data is nearly identical to that of the fully sampled data, as one would expect from the high SNRs shown in Table (2.1).

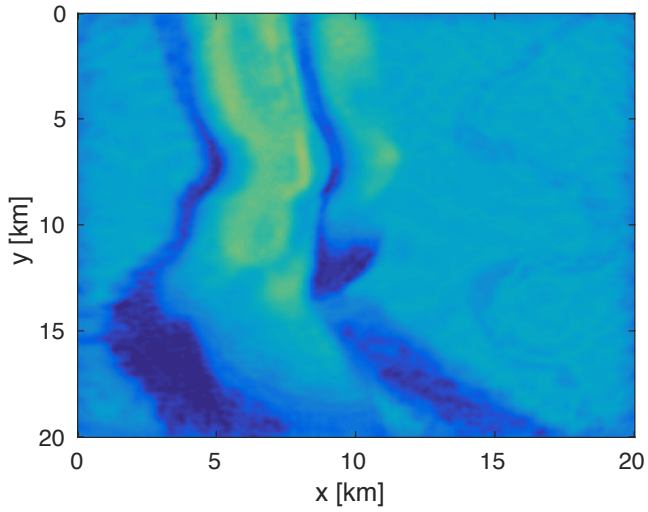


(a)

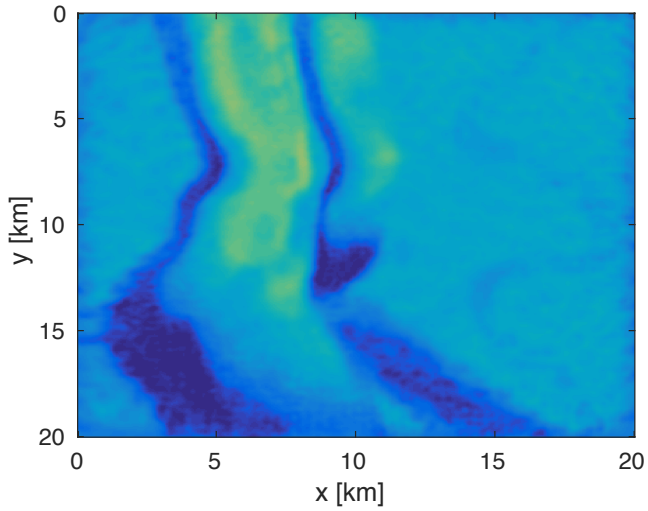


(b)

Figure 3.2: 3D Overthrust velocity model for $z = 1000m$ depth slice. (a) True model, and (b) Initial model.

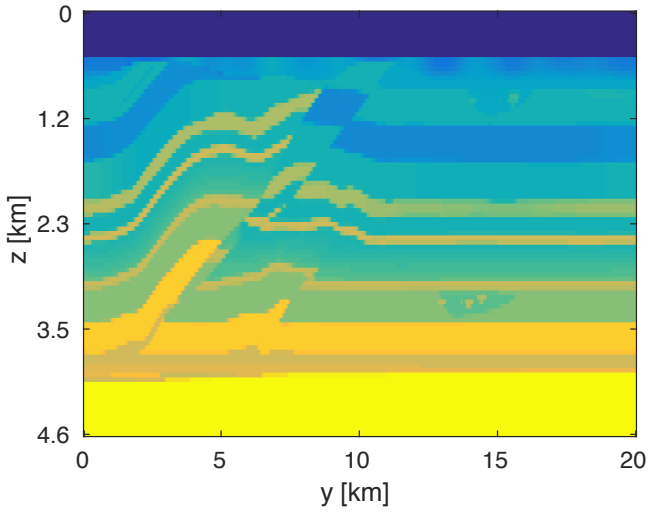


(a)

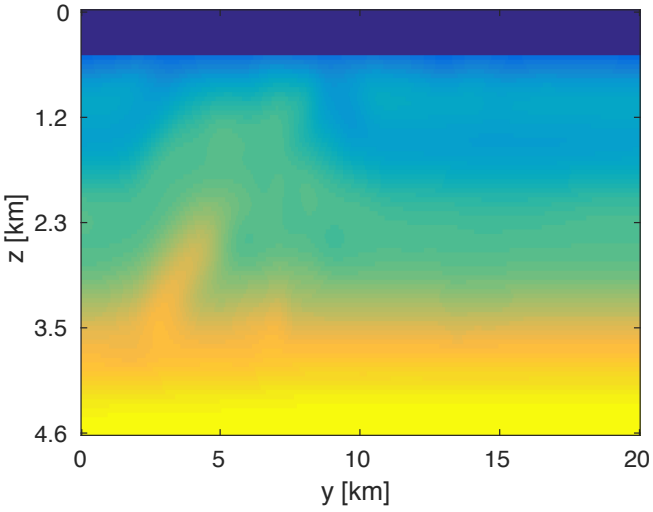


(b)

Figure 3.3: FWI results for $z = 1000m$ depth slice. (a) Inverted model with full data, and (b) Inverted model with compressed data. The number of PDE solves in both cases are the same.

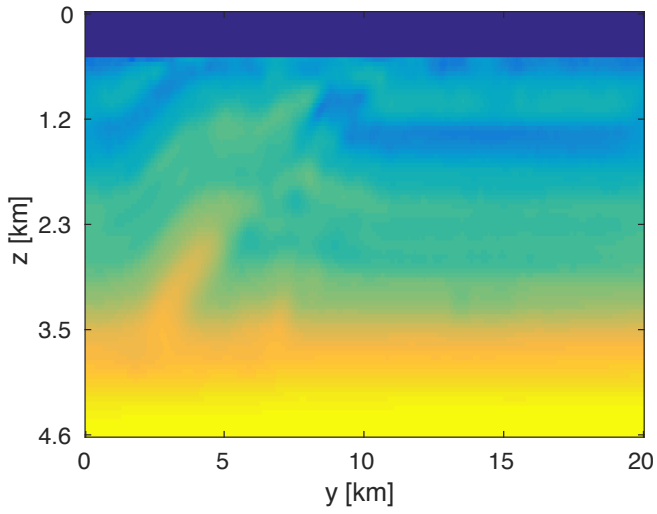


(a)

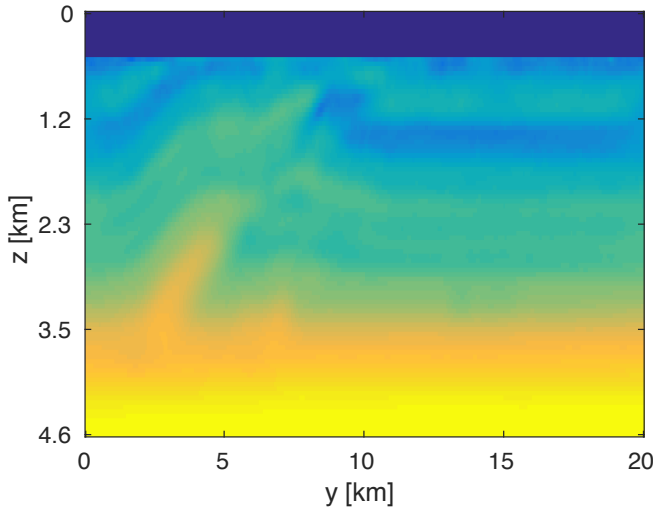


(b)

Figure 3.4: 3D Overthrust velocity model for $x = 12500m$ lateral slice. (a) True model, and (b) Initial model.

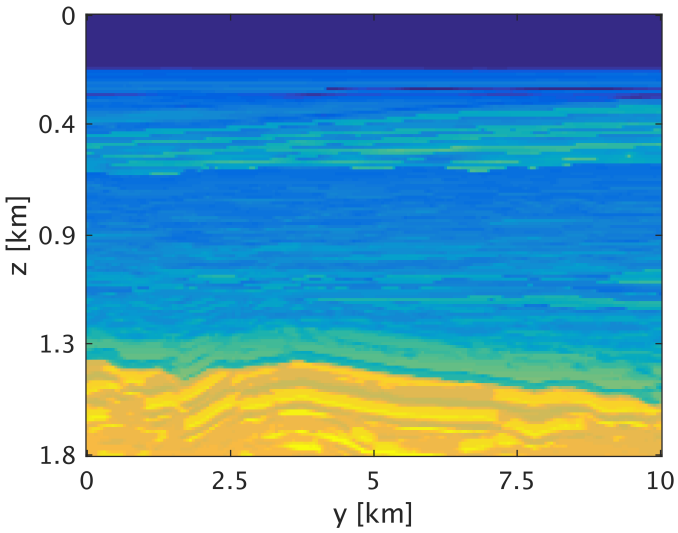


(a)

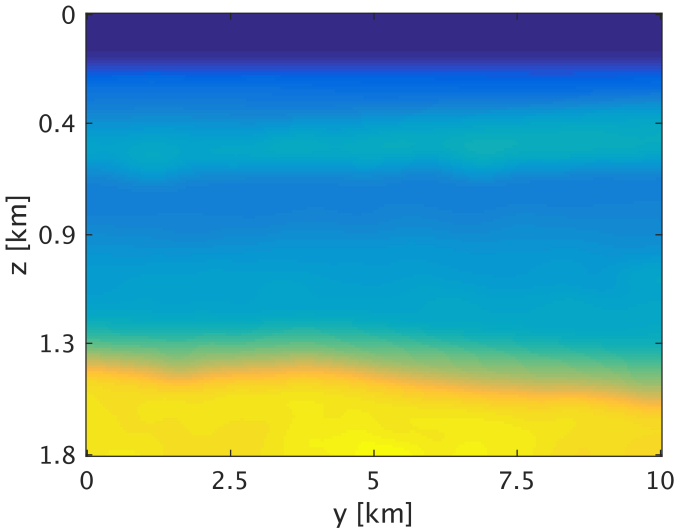


(b)

Figure 3.5: FWI results for $x = 12500m$ lateral slice. (a) Inverted model with full data, and (b) Inverted model with compressed data. The number of PDE solves in both cases are the same.

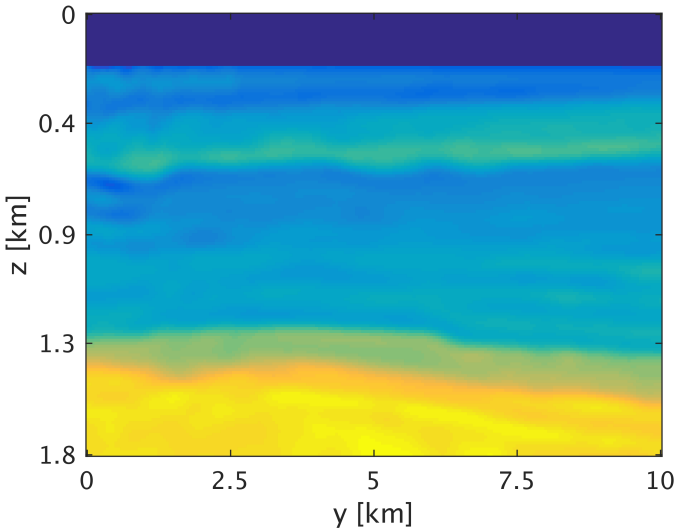


(a)

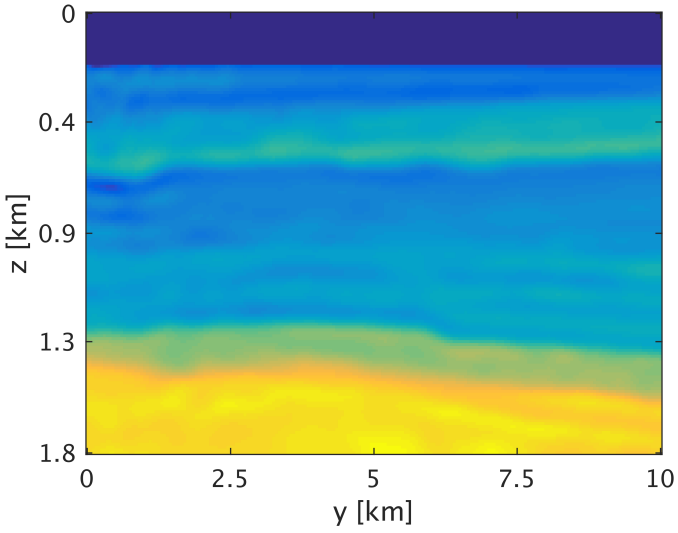


(b)

Figure 3.6: 2D slices of 3D BG velocity model along $x = 4900m$. (a) True model, and (b) Initial model.

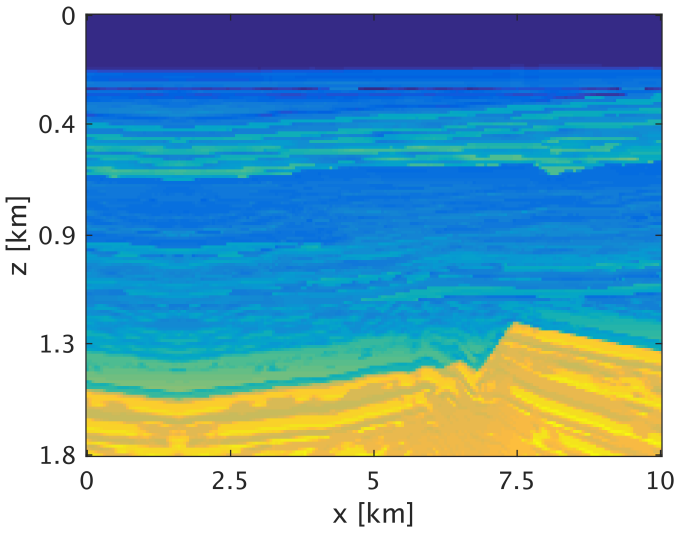


(a)

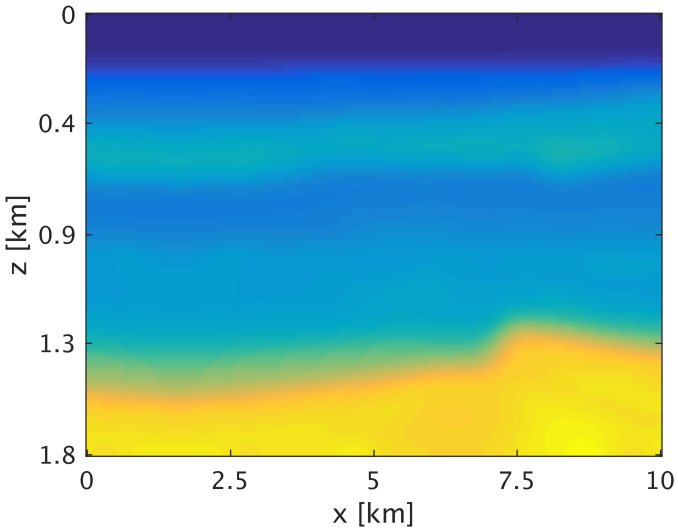


(b)

Figure 3.7: FWI results along x direction at $4900m$. (a) Inverted model with full data, and (b) Inverted model with compressed data. The number of PDE solves in both cases are the same

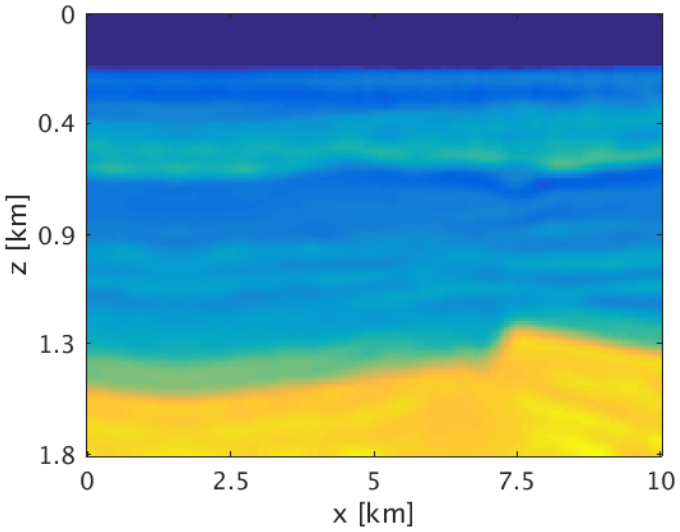


(a)

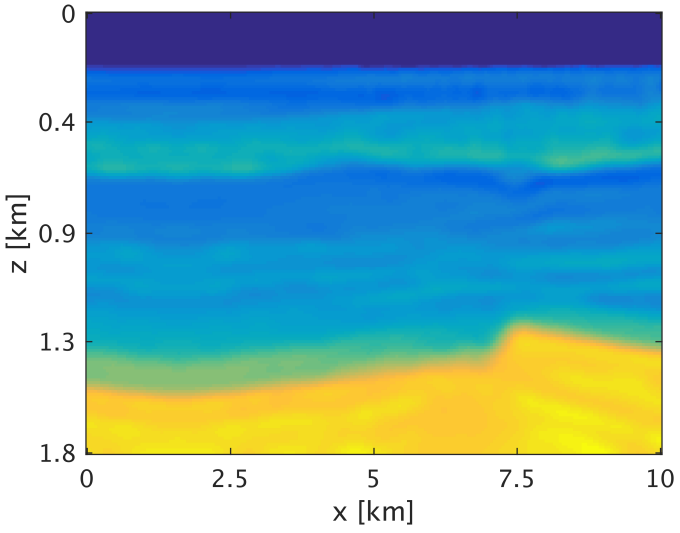


(b)

Figure 3.8: 2D slices of 3D BG velocity model along $y = 5650m$. (a) True model, and (b) Initial model.

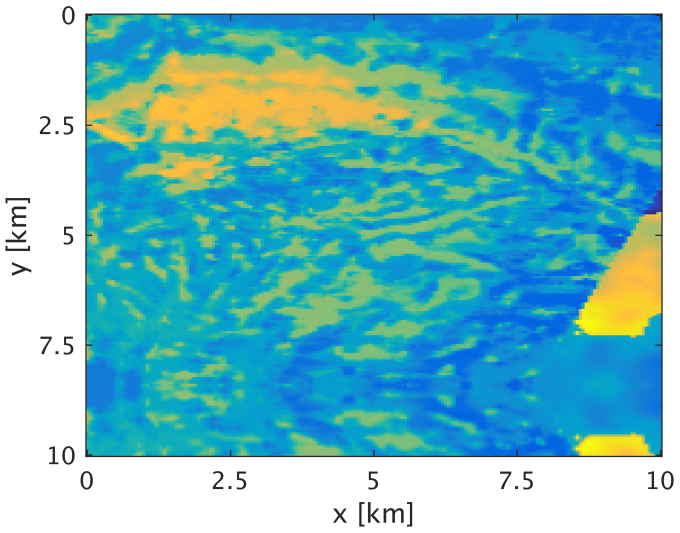


(a)

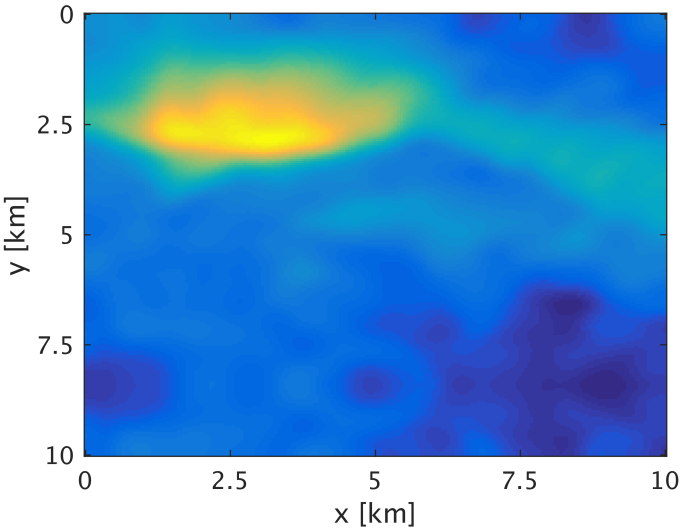


(b)

Figure 3.9: FWI results along y direction at $5650m$. (a) Inverted model with full data, and (b) Inverted model with compressed data. The number of PDE solves in both cases are the same

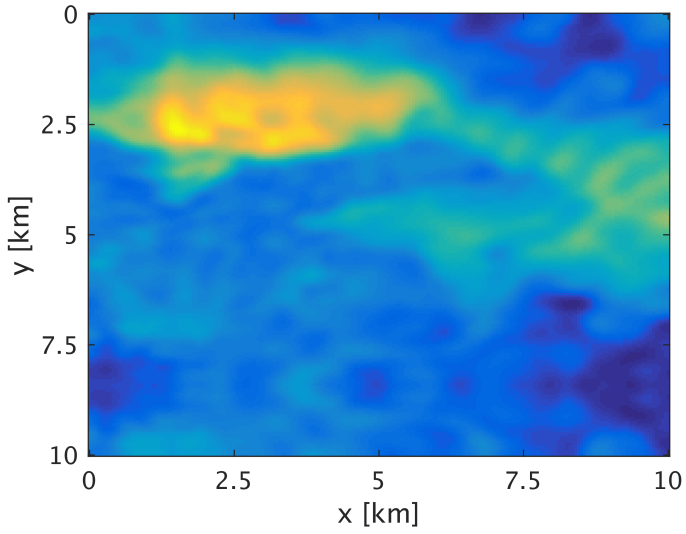


(a)

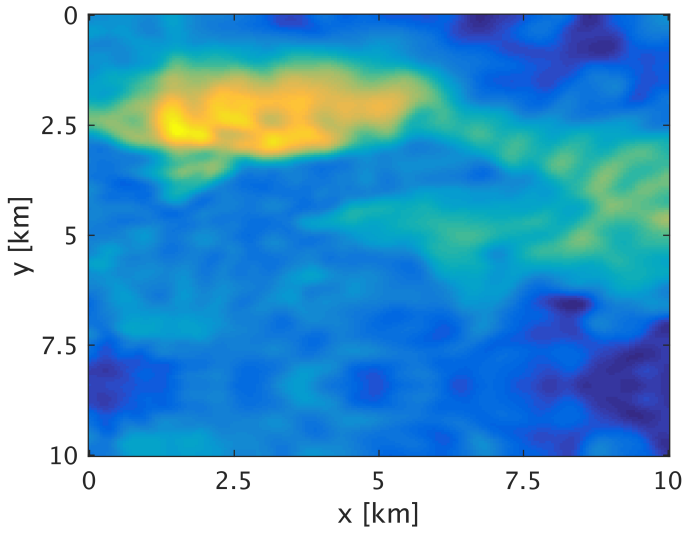


(b)

Figure 3.10: 2D slices of 3D BG velocity model along $y = 5650m$. (a) True model, and (b) Initial model.



(a)

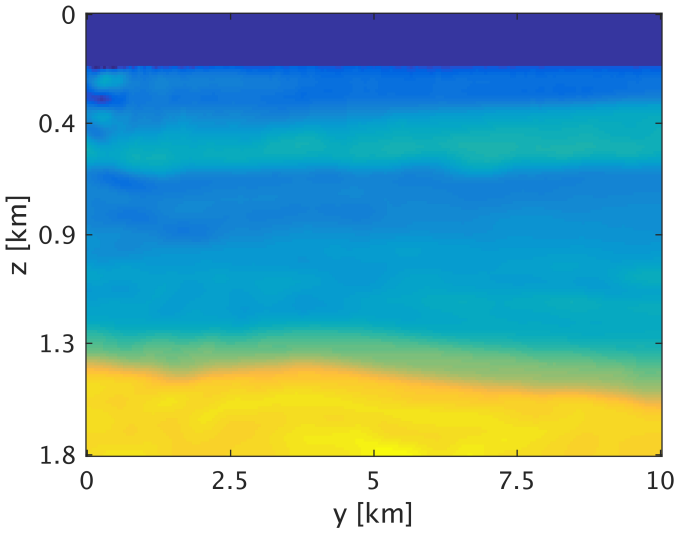


(b)

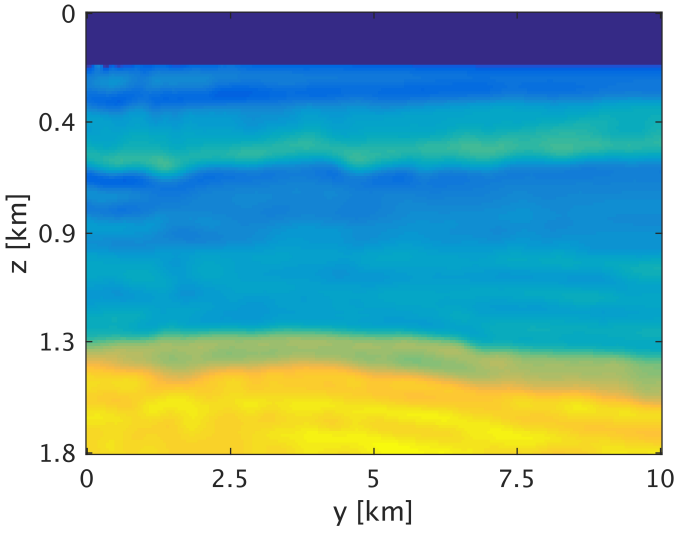
Figure 3.11: FWI results along z direction at $1200m$. (a) Inverted model with full data, and (b) Inverted model with compressed data. The number of PDE solves in both cases are the same

3.4.2 Subsampled data vs. Interpolated data

In the "missing data" scenario for our data, we remove data from our volume by randomly decimating 75% of the receiver coordinates. This is motivated by the fact that matrix/tensor completion techniques work best when the dimensions that are missing entries are well-sampled on the original grid. We consider two further inversion scenarios, one where we use the subsampled data directly in our FWI method and one where we use interpolate the data via the method in (Da Silva and Herrmann, 2015) prior to inversion. There is still an open question in the literature as to whether one should use the subsampled data directly in inversion or use interpolation. Although we do not aim to provide a comprehensive answer to this question in this work, the results in Figure 3.12 - Figure 3.14 would seem to indicate that there is a substantial benefit in interpolating the data volume prior to inversion. Compared to the results in Figure 3.6 - Figure 3.10, the inverted models arising from interpolated data are visually similar to those generated from the fully sampled data. Using the subsampled data directly, on the other hand, results in an updated model that has only been marginally updated from the initial model.

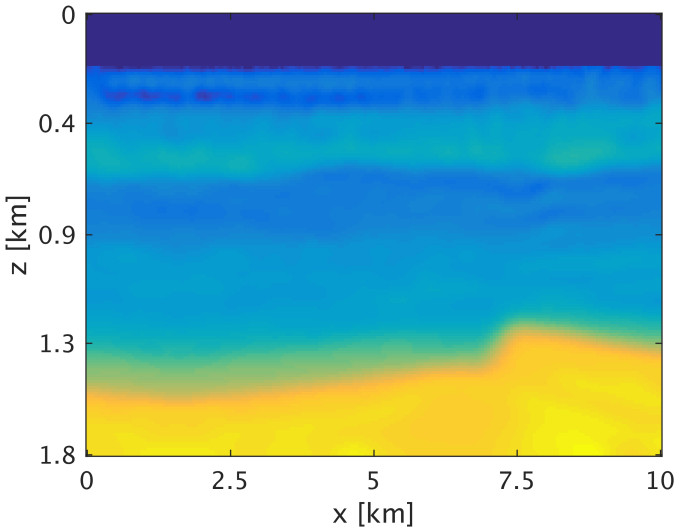


(a)

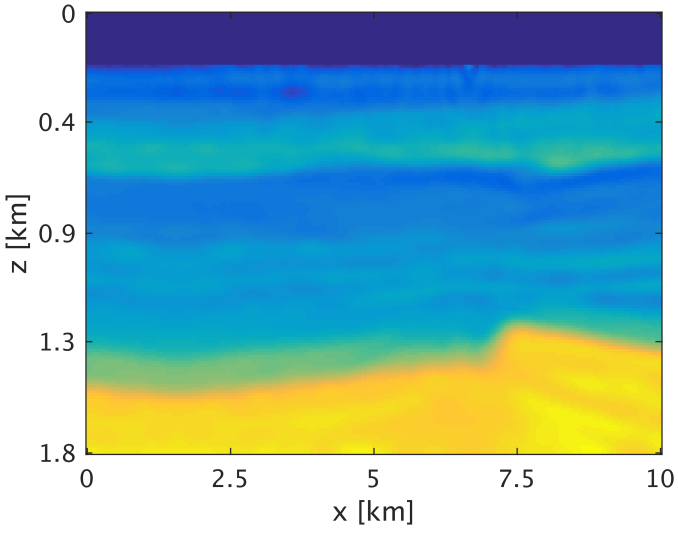


(b)

Figure 3.12: FWI results along x direction at $4900m$. (a) Inverted model with subsampled data, and (b) Inverted model with interpolated data. The number of PDE solves in both cases are the same

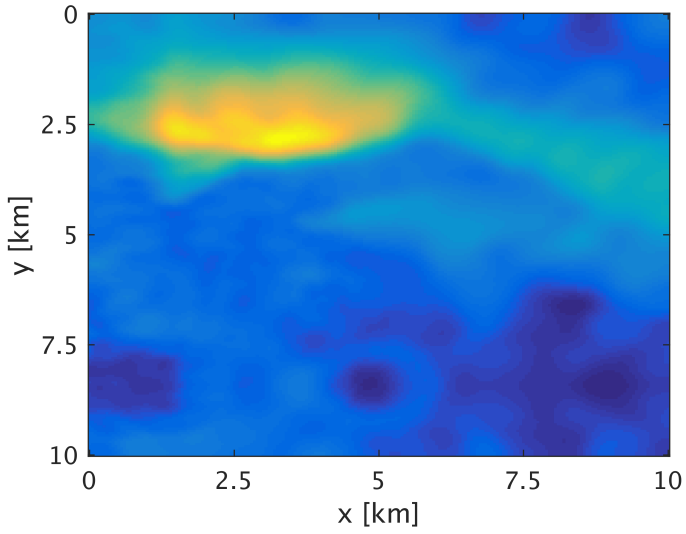


(a)

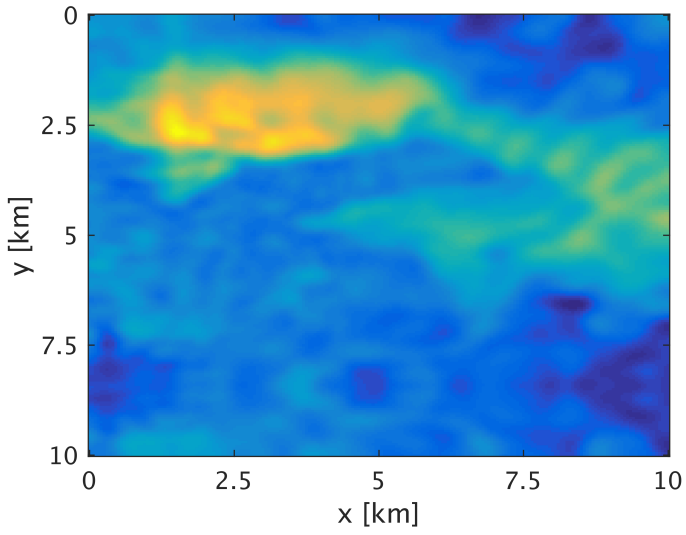


(b)

Figure 3.13: FWI results along y direction at $5650m$. (a) Inverted model with subsampled data, and (b) Inverted model with interpolated data. The number of PDE solves in both cases are the same



(a)



(b)

Figure 3.14: FWI results along z direction at $1200m$. (a) Inverted model with subsampled data, and (b) Inverted model with interpolated data. The number of PDE solves in both cases are the same

Chapter 4

Extended image volume with compressed data via probing

Image gathers are a useful method to map reflection events present in prestack seismic data to their associated reflectors in the subsurface. The end result is a function of the original spatial coordinates and a set of redundant auxiliary coordinates, the so-called *extended image*. This approach is helpful when studying angle-dependent reflection coefficients and additionally the failure of such gathers to focus can indicate errors in the large-scale background velocity model (Claerbout (1970); Doherty and Claerbout (1974); De Bruin et al. (1990); Biondi and Sava (1999); Biondi and Symes (2004); Sava and Vasconcelos (2011); Koren and Ravve (2011)). The main computational costs associated with forming these image gathers are the result of having to compute the solution of a forward and adjoint wave equation for each shot and the subsequent cost of cross-correlations. The computational costs scale linearly with the number of sources which, as the extended image itself is multi-dimensional, renders forming the full image volume computationally intractable even for 2D examples.

Traditional approaches to forming image gathers restrict the auxiliary coordinates in some fashion, such as using only surface or horizontal offsets, in order to mitigate some of the computational complexity. If computational costs were not an issue, our goal would be to have access to the information stored in the entire extended image volume to be used, for instance, in computing amplitude-versus-angle (AVA) or amplitude-versus-offset (AVO) methods derived from the linearized Zoeppritz equations (Aki and Richards (2002)).

4.1 Probing the extended image volume

In order to overcome these computational bottlenecks, the authors in Leeuwen et al. (2017) propose to *probe* the extended image volume with particular test vector $\mathbf{w}$, rather than restrict it in some ad-hoc fashion such as to a horizontal or surface offset. This approach has the advantage of not artificially limiting the information present in the extended image for computational purposes and the costs of explicit images of the subsurface scale linearly with the number of images formed, rather than the number of sources and receivers in the traditional case. We outline this approach below and demonstrate how our data extraction approach can be integrated to further reduce the memory costs of this method.

In its discretized form, the extended image $\mathbf{E}$ (at a single frequency) is a $N \times N$ complex-valued matrix, where $N = n_z n_x$ in 2D or $N = n_x n_y n_z$ in 3D. Given a background model m_0, $\mathbf{E}$ is formed as an outer product of the $N \times n_{\text{src}}$ matrices

$$\mathbf{E} = \mathbf{U}\mathbf{V}^*. \tag{4.1}$$

Here $\mathbf{U}$ and $\mathbf{V}$ are the source and receiver wavefields, respectively, which obey the equations

$$\begin{aligned} \mathbf{H}(\mathbf{m}_0)\mathbf{U} &= \mathbf{P_s}^*\mathbf{Q} \\ \mathbf{H}(\mathbf{m}_0)^*\mathbf{V} &= \mathbf{P_r}^*\mathbf{D}. \end{aligned} \tag{4.2}$$

We write $\mathbf{H}(\mathbf{m})$ to denote the discretization of the monochromatic constant-density Helmholtz equation $\nabla^2 + \omega^2\mathbf{m}(\mathbf{x})$, which is a matrix of size $N \times N$ where $N = n_z n_x$ in 2D and $N = n_x n_y n_z$ in 3D. Here $\mathbf{m}$ is the slowness squared, ∇^2 is the Laplacian, and ω is the temporal frequency in radians. The operators $\mathbf{P_s}, \mathbf{P_r}$ map the computational domain to the source and receiver locations, respectively. The adjoint of these operators inject their inputs at their respective locations into the computational domain. The data volume $\mathbf{D}$ at the current frequency contains reflection data (i.e., each column is a shot gather after removal of the direct arrival) organized as a $n_{\text{rec}} \times n_{\text{src}}$ matrix. $\mathbf{Q}$ is a $n_{\text{src}} \times n_{\text{src}}$ matrix containing the source weights associated to each shot. The full extended image is merely a sum over the extended images computed at each frequency, which we omit for notational simplicity.

Rather then forming this volume explicitly, which would require the solution of $n_{\text{src}} n_{\text{rec}} n_{\text{freq}}$ PDEs, the work of Leeuwen et al. (2017) introduced the notion of *matrix probing* in this context, which is to say selecting a

column of $\mathbf{E}$ implicitly by multiplying it with a vector $\mathbf{w}$. The product $\mathbf{y} = \mathbf{E}\mathbf{w}$ can be computed as

$$\mathbf{y} = \mathbf{H}^{-1}\mathbf{P_s}^{*}\mathbf{Q}\mathbf{D}^{*}\mathbf{P_r}\mathbf{H}^{-1}\mathbf{w} \tag{4.3}$$

at the cost of two PDE solves. When $\mathbf{w} = [0, \ldots, 0, 1, 0, \ldots, 0]$ is an identity basis vector, $\mathbf{E}\mathbf{w}$ is a common-image-point at the location (x_k, y_k, z_k) associated to the index of the non-zero entry. Other choices of probing vector s can be used for purposes other than imaging, such as using Gaussian vector s in the context of Wave Equation Migration Velocity Analysis (WEMVA).

From Leeuwen et al. (2017), forming $\mathbf{E}\mathbf{w}$ involves computing the quantities in (4.3), read from right to left. We can take advantage of keeping our data volume in compressed form throughout this computation by Algorithm 4.1, which involves the computation of the term $\mathbf{D}^{*}\mathbf{P_r}\mathbf{H}^{-1}\mathbf{w}$, which we write as $\mathbf{D}^{*}\mathbf{v}$. This algorithm is simply a matrix-vector multiplication with the explicit matrix row extraction replaced with Algorithm 2.1. The overhead introduced by this method is minimal but the memory savings are significant.

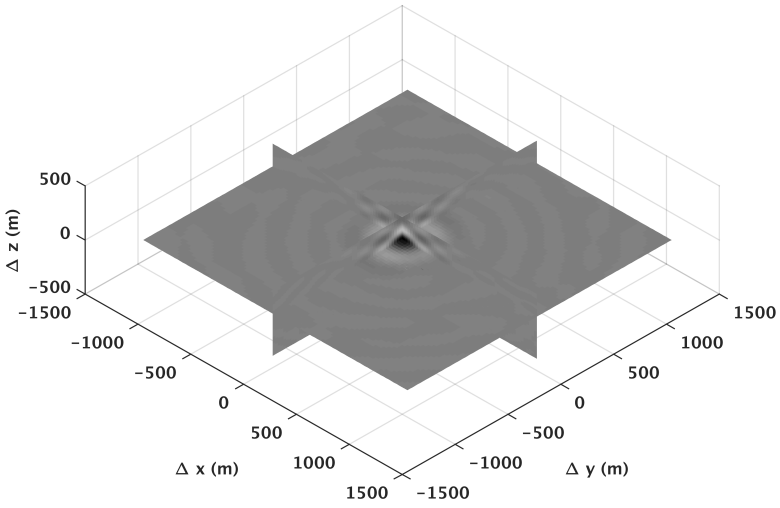
Algorithm 4.1 Forming the CIP gathers with compressed Hierarchical Tucker parameters

Input: $\mathbf{v} = P_r\mathbf{H}^{-1}\mathbf{w}$
Output: $\mathbf{z} = \mathbf{D}^{*}\mathbf{v}$
For each source index $\mathbf{i} = (i_{\mathrm{src}_x}, i_{\mathrm{src}_y})$
1. Extract the common source gather from the data using Algorithm 2.1, resulting in $\mathbf{D_i}$.
2. Correlate $\overline{\mathbf{D_i}}$ and $\mathbf{v}$ to produce $\mathbf{z_i}$.

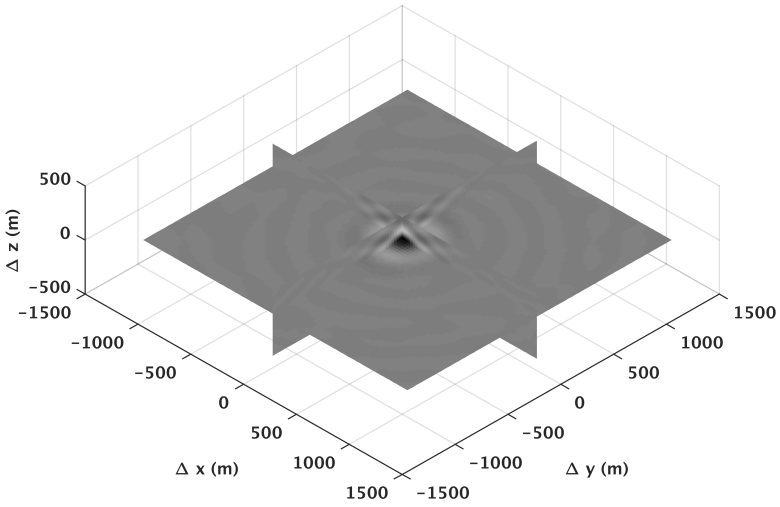
4.2 Performance results

We showcase our data compression and extraction method integrated in to the context of forming common-image-point (CIP) gathers. Our model is a 2.5 km $\times$ 2.5 km $\times$ 0.6 km subset of the BG Compass model with 25m x 25m x 6m spacing, resulting in a model grid of size 101^3. We generate data using 15 frequencies ranging from 5 to 12 Hz with 0.5 Hz spacing and a Ricker wavelet with central frequency of 15Hz. The sources are placed at the surface of the ocean with 75 m spacing and the receivers are located on the sea floor every 50 m, resulting in 1156 sources and 2601 receivers. We

compare forming the CIP with the true data as well as the volume in which each monochromatic frequency slice is truncated in the HT format with 1% relative error. Figure 4.1 shows the 3D CIP using both data volumes and Figure 4.2 - Figure 4.4 plots various 2D slices through the computed volume, extracted at (x, y, z) coordinates $(1250m, 1250m, 390m)$. Given the compressibility of seismic data in HT format, the plots using each data set are virtually indistinguishable and, using compressed data, the CIP gather has an SNR of 49.8 dB.

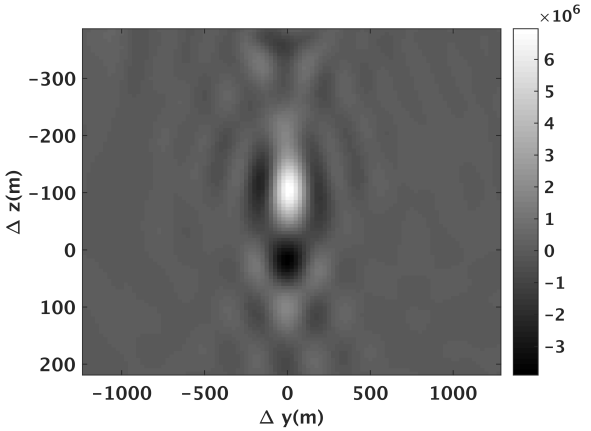


(a)

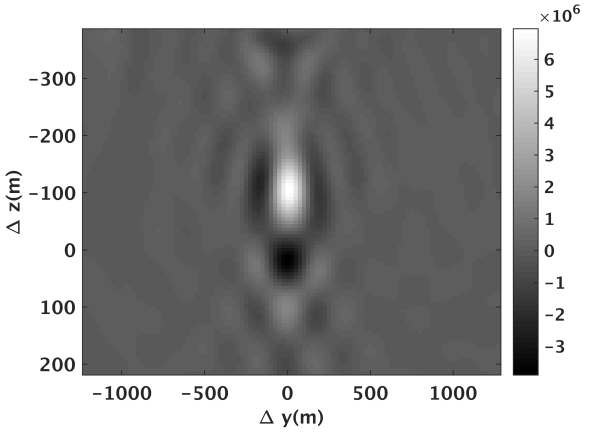


(b)

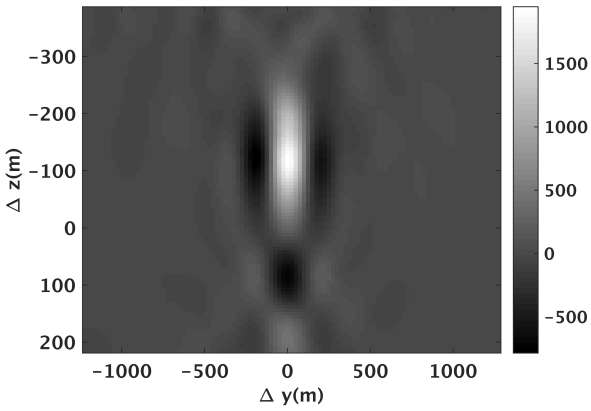
Figure 4.1: A full CIP gather extracted at location (x, y, z) = (1250m, 1250m, 390m). (a) with fully sampled data, and (b) with compressed HT parameters.



(a)

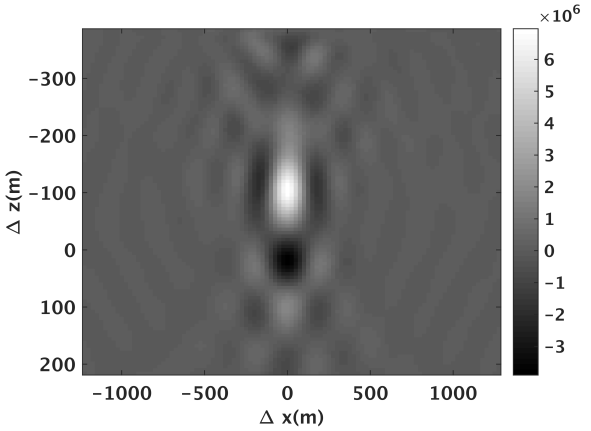


(b)

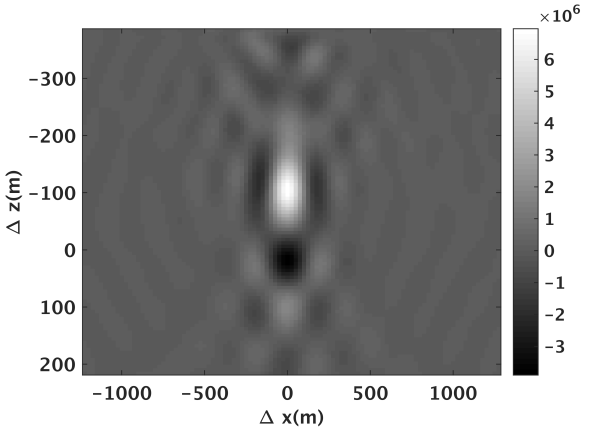


(c)

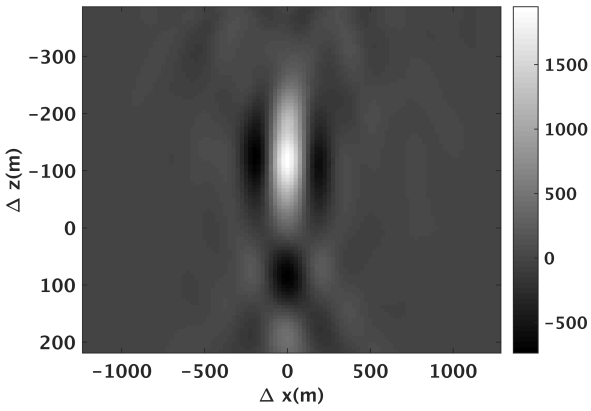
Figure 4.2: 2D slices extracted from the CIP gathers in Figure 4.1 along horizontal offset. (a) is computed using the original data, (b) uses HT compressed data, and (c) displays the difference between the above two on a colorbar 100x smaller than the results



(a)

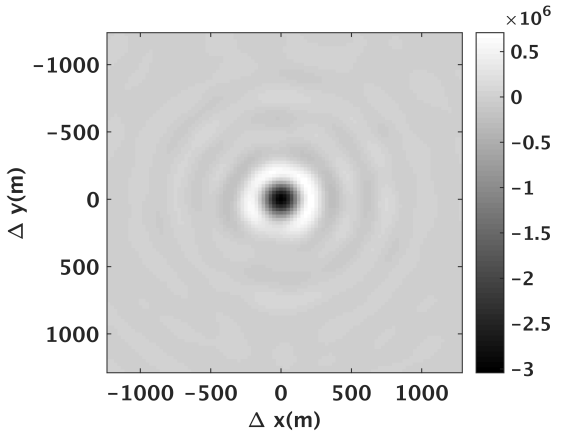


(b)

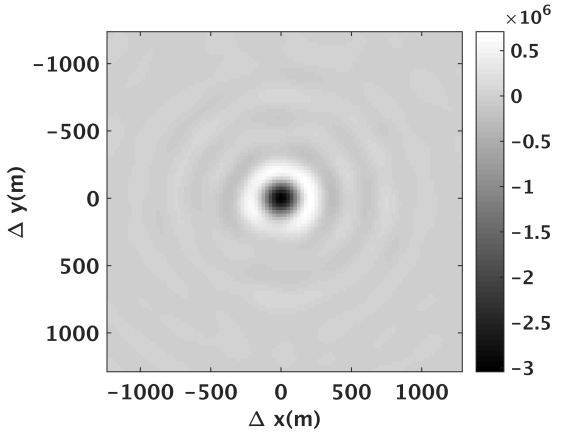


(c)

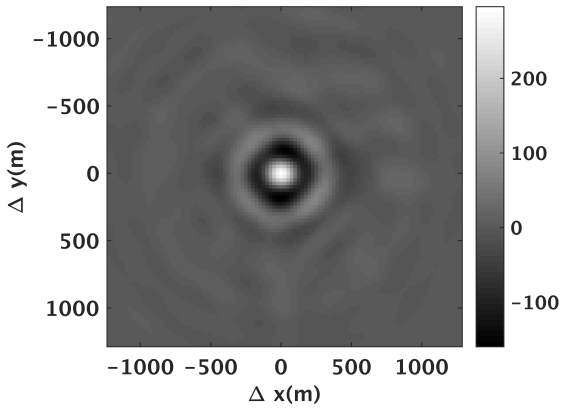
Figure 4.3: 2D slices extracted from the CIP gathers in Figure 4.1 along another horizontal offset. (a) is computed using the original data, (b) uses HT compressed data, and (c) displays the difference between the above two on a colorbar 100x smaller than the results



(a)



(b)



(c)

Figure 4.4: 2D slices extracted from the CIP gathers in Figure 4.1 along vertical offset. (a) is computed using the original data, (b) uses HT compressed data, and (c) displays the difference between the above two on a colorbar 100x smaller than the results

Chapter 5

Discussion and Conclusion

Working with the data directly in a compressed form has the potential to significantly reduce the memory and communication costs associated with storing and distributing the data volume in a large parallel environment. In particular in a cloud-based computing environment, being able to avoid cross-node communication costs associated with transmitted subsets of the data is important for the scaling of these algorithms to thousands of cores. Our approach also offers the possibility of moving towards an asynchronous full waveform paradigm, where each node has a local copy of the model and only synchronizes its copy periodically with its neighbours, while still having access to a full copy of the compressed data locally. This is an exciting and pertinent topic for future research.

One of the key issues in low-rank compression and interpolation is that the original signal must be sufficiently *compressible* i.e., well approximated by a low-rank tensor. In practical terms, this requires that the singular values of the data, when matricized along the appropriate dimensions, must decay quickly in order for this low-rank approach to work. As shown in (Aravkin et al. (2014); Da Silva and Herrmann (2013); Kumar et al. (2015)), necessary conditions for fast singular value decay are considering the data in an appropriate *transform domain* as well as having *sufficient spatial sampling* of the tensor. One of the challenges for low-rank methods in seismic data is that the effective ranks of the data matricizations increase as the temporal frequency of the data, rendering them less favourable to approximation by a low-rank tensor. The examples considered in this work have therefore been restricted to a relatively low frequency range. Potentially appropriate multidimensional windowing can alleviate this shortcoming (Kumar et al. (2013)), but this remains to be seen in the 3D seismic case. Additionally, as the error level for the truncation approach is user-defined, we are able to produce inversion and image gather results that differ only slightly from using the uncompressed data volume. When the input data is noisy, the HT format can also reduce the imprint of the noise, even in high-amplitude

scenarios Da Silva and Herrmann (2014).

Low-rank methods are an efficient approach to dealing with the curse of dimensionality. In addition to being simple to implement and efficient from a memory perspective, they offer an opportunity to compute quantities directly in their compressed form. Using the Hierarchical Tucker format allows us to drastically reduce the storage costs of seismic data while offering us the flexibility to extract shot gathers from the corresponding HT parameters. When we have fully sampled data, we can use existing truncation techniques to compress our data in to HT form. The subsequent inversion results using both fully sampled and compressed data differ only slightly from each other, while the memory savings for the data are substantial. When the data has missing entries, we see a significant improvement in the inversion results from the HT interpolated data compared to merely using the subsampled data. In the image volume context, we were able to seamlessly integrate our approach in to forming image gathers through matrix-probing and the results are again only marginally affected by the use of compressed data. This method has the potential to drastically reduce communication costs in large-scale distributed environments, which will be particularly relevant as seismic data grows from the petabyte to exabyte scales.

Bibliography

Abma, R., Kelley, C., Kaldy, J., et al. (2007). Sources and treatments of migration-introduced artifacts and noise. In *2007 SEG Annual Meeting*. Society of Exploration Geophysicists. 2

Aki, K. and Richards, P. G. (2002). *Quantitative seismology*. Freeman and Co. New York. 45

Aravkin, A., Kumar, R., Mansour, H., Recht, B., and Herrmann, F. J. (2014). Fast methods for denoising matrix completion formulations, with applications to robust seismic data interpolation. *SIAM Journal on Scientific Computing*, 36(5):S237–S266. 10, 15, 53

Bekara, M. and Van der Baan, M. (2007). Local singular value decomposition for signal enhancement of seismic data. *Geophysics*, 72(2):V59–V65. 10

Berkhout, A. J. (2012). *Seismic migration: Imaging of acoustic energy by wave field extrapolation*, volume 12. Elsevier. 2

Biondi, B. and Almomin, A. (2014). Simultaneous inversion of full data bandwidth by tomographic full-waveform inversion. *Geophysics*, 79(3):WA129–WA140. 26

Biondi, B. and Sava, P. (1999). Wave-equation migration velocity analysis. In *SEG Technical Program Expanded Abstracts 1999*, pages 1723–1726. Society of Exploration Geophysicists. 45

Biondi, B. and Symes, W. W. (2004). Angle-domain common-image gathers for migration velocity analysis by wavefield-continuation imaging. *Geophysics*, 69(5):1283–1298. 45

Bording, R. P., Gersztenkorn, A., Lines, L. R., Scales, J. A., and Treitel, S. (1987). Applications of seismic travel-time tomography. *Geophysical Journal International*, 90(2):285–303. 26

Candès, E., Demanet, L., Donoho, D., and Ying, L. (2006a). Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3):861–899. 10

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772. 10

Candès, E. J., Romberg, J., and Tao, T. (2006b). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509. 5

Candès, E. J. and Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080. 10

Carroll, J. D. and Chang, J.-J. (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319. 11

Cheng, J. and Sacchi, M. D. (2015). Separation and reconstruction of simultaneous source data via iterative rank reduction. *Geophysics*, 80(4):V57–V66. 10

Claerbout, J. F. (1970). Coarse grid calculations of waves in inhomogeneous media with application to delineation of complicated seismic structure. *Geophysics*, 35(3):407–418. 45

Curry, W. (2010). Interpolation with Fourier-radial adaptive thresholding. *Geophysics*, 75(6):WB95–WB102. 11

Da Silva, C. and Herrmann, F. (2013). Hierarchical Tucker tensor optimization-applications to 4D seismic data interpolation. In *75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013*. ix, 14, 53

Da Silva, C. and Herrmann, F. J. (2014). Low-rank promoting transformations and tensor interpolation-applications to seismic data denoising. In *76th EAGE Conference and Exhibition 2014*. 54

Da Silva, C. and Herrmann, F. J. (2015). Optimization on the hierarchical Tucker manifold–Applications to tensor completion. *Linear Algebra and its Applications*, 481:131–173. ix, 6, 7, 10, 11, 13, 15, 16, 18, 41

Da Silva, C. and Herrmann, F. J. (2016). A unified 2D/3D software environment for large scale nonlinear inverse problems. *In Preparation*. 27, 30

De Bruin, C., Wapenaar, C., and Berkhout, A. (1990). Angle-dependent reflectivity by means of prestack migration. *Geophysics*, 55(9):1223–1234. 45

De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278. 11

Demanet, L. (2006). *Curvelets, Wave Atoms, and Wave Equations*. PhD thesis, California Institute of Technology. 15

Doherty, S. and Claerbout, J. (1974). Velocity analysis based on the wave equation. *Geophysics*, 37:741–768. 45

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306. 5

Freire, S. L. and Ulrych, T. J. (1988). Application of singular value decomposition to vertical seismic profiling. *Geophysics*, 53(6):778–785. 10

Gower, R., Goldfarb, D., and Richtárik, P. (2016). Stochastic block BFGS: squeezing more curvature out of data. In *International Conference on Machine Learning*, pages 1869–1878. 28

Grasedyck, L. (2010). Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054. 6, 7, 13, 19

Gray, S. H. (2013). Spatial sampling, migration aliasing, and migrated amplitudes. *Geophysics*, 78(3):S157–S164. 2

Hackbusch, W. and Kühn, S. (2009). A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722. 6, 11

Hennenfent, G. and Herrmann, F. J. (2006). Application of stable signal recovery to seismic data interpolation. In *SEG Technical Program Expanded Abstracts 2006*, pages 2797–2801. Society of Exploration Geophysicists. 5, 11

Herrmann, F. J. and Hennenfent, G. (2008). Non-parametric seismic data recovery with curvelet frames. *Geophysical Journal International*, 173(1):233–248. 5, 10, 11

Herrmann, F. J., Wang, D., Hennenfent, G., and Moghaddam, P. P. (2007). Curvelet-based seismic data processing: A multiscale and nonlinear approach. *Geophysics*, 73(1):A1–A5. 10

Jones, I. F. and Davison, I. (2014). Seismic imaging in and around salt bodies. *Interpretation*, 2(4):SL1–SL20. 2

Jumah, B. and Herrmann, F. J. (2014). Dimensionality-reduced estimation of primaries by sparse inversion. *Geophysical Prospecting*, 62(5):972–993. 10

Kabir, M. N. and Verschuur, D. Restoration of missing offsets by parabolic Radon transform. *Geophysical Prospecting*, 43(3):347–368. 11

Koren, Z. and Ravve, I. (2011). Full-azimuth subsurface angle domain wavefield decomposition and imaging Part i: Directional and reflection image gathers. *Geophysics*. 45

Krebs, J. R., Anderson, J. E., Hinkley, D., Neelamani, R., Lee, S., Baumstein, A., and Lacasse, M.-D. (2009). Fast full-wavefield seismic inversion using encoded sources. *Geophysics*, 74(6):WCC177–WCC188. 26

Kreimer, N. and Sacchi, M. D. (2012). A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation. *Geophysics*, 77(3):V113–V122. 6, 10, 11

Kreimer, N., Stanton, A., and Sacchi, M. D. (2013). Tensor completion based on nuclear norm minimization for 5D seismic data reconstruction. *Geophysics*, 78(6):V273–V284. 10

Kumar, R., Da Silva, C., Akalin, O., Aravkin, A. Y., Mansour, H., Recht, B., and Herrmann, F. J. (2015). Efficient matrix completion for seismic data reconstruction. *Geophysics*, 80(5):V97–V114. 6, 10, 11, 15, 53

Kumar, R., Mansour, H., Herrmann, F. J., and Aravkin, A. Y. (2013). Reconstruction of seismic wavefields via low-rank matrix factorization in the hierarchical-separable matrix representation. In *SEG Technical Program Expanded Abstracts 2013*, pages 3628–3633. Society of Exploration Geophysicists. 53

Kumar, R., Sharan, S., Wason, H., and Herrmann, F. (2016). Time-jittered marine acquisition: A rank-minimization approach for 5D source separation. In *SEG Technical Program Expanded Abstracts 2016*, pages 119–123. Society of Exploration Geophysicists. 10

Leeuwen, T., Kumar, R., and Herrmann, F. J. (2017). Enabling affordable omnidirectional subsurface extended image volumes via probing. *Geophysical Prospecting*, 65(2):385–406. 46, 47

Leveille, J. P., Jones, I. F., Zhou, Z.-Z., Wang, B., and Liu, F. (2011). Subsalt imaging for exploration, production, and development: A review. *Geophysics*, 76(5):WB3–WB20. 2

Li, X., Aravkin, A. Y., van Leeuwen, T., and Herrmann, F. J. (2012). Fast randomized full-waveform inversion with compressive sensing. *Geophysics*, 77(3):A13–A17. 26

Ma, J. (2013). Three-dimensional irregular seismic data reconstruction via low-rank matrix completion. *Geophysics*, 78(5):V181–V192. 10

Moritz, P., Nishihara, R., and Jordan, M. (2016). A linearly-convergent stochastic l-bfgs algorithm. In *Artificial Intelligence and Statistics*, pages 249–258. 28

Mothi, S., Schwarz, K., and Zhu, H. (2013). Impact of full-azimuth and long-offset acquisition on Full Waveform Inversion in deep water Gulf of Mexico. In *75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013*. 2

Nazari Siahsar, M. A., Gholtashi, S., Kahoo, A. R., Marvi, H., and Ahmadifard, A. (2016). Sparse time-frequency representation for seismic noise reduction using low-rank and sparse decomposition. *Geophysics*, 81(2):V117–V124. 10

Oropeza, V. and Sacchi, M. (2011). Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis. *Geophysics*, 76(3):V25–V32. 11

Pratt, R. G. (1999). Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model. *Geophysics*, 64(3):888–901. 26

Recht, B. (2011). A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec):3413–3430. 10

Sacchi, M., Kaplan, S., and Naghizadeh, M. (2009). Fx gabor seismic data reconstruction. In *71st EAGE Conference and Exhibition incorporating SPE EUROPEC 2009*. 11

Saragiotis, C., Alkhalifah, T., and Fomel, S. (2013). Automatic traveltime picking using instantaneous traveltime. *Geophysics*, 78(2):T53–T58. 26

Sava, P. and Vasconcelos, I. (2011). Extended imaging conditions for wave-equation migration. *Geophysical Prospecting*, 59(1):35–55. 45

Schmidt, M. W., Van Den Berg, E., Friedlander, M. P., and Murphy, K. P. (2009). Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm. In *12th International Conference on Artificial Intelligence and Statistics*. 28

Schuster, G. T. (2007). Basics of seismic wave theory. *University of Utah*. 26

Sheriff, R. E. and Geldart, L. P. (1995). *Exploration seismology*. Cambridge university press. 1

Stork, C. (1992). Singular value decomposition of the velocity-reflector depth tradeoff, Part 2: High-resolution analysis of a generic model. *Geophysics*, 57(7):933–943. 10

Taillandier, C., Noble, M., Chauris, H., and Calandra, H. (2009). First-arrival traveltime tomography based on the adjoint-state method. *Geophysics*, 74(6):WCB1–WCB10. 26

Tarantola, A. (1987). Inverse Problem Theory Elsevier. *New York*. 26

Tobler, C. (2012). *Low-rank tensor methods for linear systems and eigenvalue problems*. PhD thesis, ETH Zürich. 16

Trickett, S., Burroughs, L., and Milton, A. (2013). Interpolation using Hankel tensor completion. *Ratio*, 1(16):16. 10, 11

Trickett, S., Burroughs, L., Milton, A., Walton, L., Dack, R., et al. (2010). Rank-reduction-based trace interpolation. In *2010 SEG Annual Meeting*. Society of Exploration Geophysicists. 10

Villasenor, J. D., Ergas, R., and Donoho, P. (1996). Seismic data compression using high-dimensional wavelet transforms. In *Data Compression Conference, 1996. DCC'96. Proceedings*, pages 396–405. IEEE. 10, 11

Virieux, J. and Operto, S. (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26. 2

Wang, J., Ng, M., and Perz, M. (2010). Seismic data interpolation by greedy local Radon transform. *Geophysics*, 75(6):WB225–WB234. 11

Yilmaz, Ö. (2001). *Seismic data analysis*, volume 1. Society of Exploration Geophysicists Tulsa. 1, 2

Ying, L., Demanet, L., and Candès, E. (2005). 3D discrete curvelet transform. In *Optics & Photonics 2005*, pages 591413–591413. International Society for Optics and Photonics. 10