UNIVERSITY
OF OSLO

Mohammed Outhmane Faouzi Zizi

Seismic Data Processing in a Compressed Domain using Constrained Dictionary Learning

Thesis submitted for the degree of Philosophiae Doctor

Department of Geosciences
Faculty of Mathematics and Natural Sciences

University of Oslo

2023

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 2634

ISSN 1501-7710

Cover: UiO.
Print production: Graphic center, University of Oslo.

Preface

This thesis is submitted to the Faculty of Mathematics and Natural Sciences at the University of Oslo (UiO) in fulfillment of the requirements for the degree of Philosophiae Doctor (Ph.D.). The work presented here was carried out in collaboration between the Acquisition Geophysics group in the Research and Development (R&D) Department of PGS and the Department of Geosciences of UiO. This collaboration was made possible through the Industrial PhD Program organized and supported by the Research Council of Norway.

The research described in this thesis was conducted between January 2020 and March 2023 under the supervision of Dr. Anthony Day, Dr. Pierre Turquais, and Dr. Morten W. Pedersen from PGS, and Prof. Leiv-Jacob Gelius from UiO. The thesis consists of three papers presented in chronological order of writing. Two of these papers have already been published in the journals *Geophysics* and *Geophysical Prospecting*, respectively, while the third paper has been submitted to *Geophysical Prospecting* and is currently under review.

Acknowledgements

I would like to express my deepest gratitude to my four supervisors, Dr. Anthony Day, Dr. Pierre Turquais, Dr. Morten W. Pedersen, and Prof. Leiv-Jacob Gelius, for their constant support, guidance, and encouragement throughout my Ph.D. journey. Dr. Day has been an exceptional supervisor, especially during the difficult times caused by the COVID-19 pandemic and company restructuring. His unwavering commitment and dedication to my research were critical to its continuation and success, and I am thankful for the opportunity to have worked under his mentorship. I am also deeply grateful to Dr. Pierre Turquais, whose technical expertise and knowledge have been invaluable to my research. He has always been available to provide technical assistance, insights on the practical aspects of the research, and his dedication and hard work have been an inspiration to me. Dr. Turquais has been a true mentor to me throughout this Ph.D. journey. Furthermore, I appreciate Dr. Morten W. Pedersen's enthusiastic participation and commitment to our discussions, which greatly enriched my research. His guidance and insight from his vast industry experience were immensely helpful, and I am grateful to have learned from him and worked with him throughout this project. Additionally, Prof. Leiv-Jacob Gelius' unwavering support and motivation have been a source of inspiration throughout my Ph.D. journey. His encouragement and confidence in my abilities helped me to grow, achieve my goals, and never give up. I would also like to acknowledge Dr. Walter Söllner and Dr. Stefan Jetschny, who served as my supervisors for the first three months of this Ph.D. project. Their contributions and guidance were greatly appreciated, and I am grateful for the foundation they helped establish for my research.

Furthermore, I received tremendous support from all the colleagues in the Research and Development (R&D) Department of PGS. Their expertise, insights, and constructive criticism were invaluable in shaping my research. I am especially thankful to my colleagues in the Acquisition Geophysics group, whose contributions to this project have been vital to its success. They provided a welcoming and stimulating work environment that made a significant difference in my research.

Lastly, I would like to express my sincere gratitude to my family, parents, and brother for their constant support, motivation, and encouragement throughout my Ph.D. journey. Without their unwavering love and guidance, I would not have been able to achieve what I have. I am also grateful to my girlfriend, Kenza Idrissi Kaitouni, for her love and belief in me. Her presence in my life has been invaluable. She has been my rock and source of strength during the ups and downs of this journey.

Mohammed Outhmane Faouzi Zizi
Oslo, May 2023

Summary

Seismic exploration surveys are increasingly covering larger areas and utilizing a greater number of sensors to collect data. Hence, the resulting seismic data sets are rapidly growing in size. However, this increase in data volume, presents a significant challenge for conventional seismic processing and imaging techniques, which require extensive human and computational resources . Compressing the seismic data at an early stage of the seismic processing sequence can be key element to overcoming storage and data transfer barriers. Moreover, applying seismic processing steps directly in the compressed domain would not only save storage and transfer costs but could also lead to faster and more cost-effective alternatives to standard seismic processing.

The relevant information contained within seismic data sets is of smaller dimensionality than the data themselves. Consequently, the seismic data can be expressed with a reduced number of coefficients compared to the number of data samples by transforming the data into an appropriate mathematical domain. Such transformations are generally called sparse and have gained more interest in seismic processing during the last few years. This thesis first explores different sparse representations in the context of seismic data compression, where a distinction is made between fixed and learned transforms. It then follows an investigation of different transforms that succeed in describing the seismic wavefield based on their analytical expressions. Further, a particular focus is placed on a parabolic constrained version of dictionary learning methods, where the seismic wavefield can be locally explained by kinematic parameters. The extracted kinematic parameters are exploited to develop new processing methods based on simple operators that are directly applied to compressed seismic data.

This goal has been tackled with a step by step approach as follows: (1) A dictionary learning (DL)-based compression method is developed, where redundancy in seismic data is fully exploited by learning small-sized dictionaries from local windows of the seismic shot gathers. Our method has been evaluated on both synthetic and realistic data sets and demonstrated superior effectiveness compared to conventional compression methods, which are based on different predefined transforms. (2) A novel method that applies the dual-sensor wavefield separation processing step in the compressed domain has been developed for 2D seismic data based on parabolic dictionary learning. Kinematic parameters such as the slope and curvature of the learned atoms are used to allow the wavefield separation processing step directly in the dictionary learning compressed domain. The method has achieved similar results as an industry-standard FK-based method for wavefield separation and has the advantage of being robust to spatial aliasing without the need for interpolation, while reaching a high compression performance. (3) A deghosting process in the compressed domain, specifically designed for low-frequencies has been developed. The method can

be seen as complementary to the previous wavefield separation method which usually handles only high frequencies. By operating directly on the compressed data format, which is of smaller size, the method has provided a cost-effective alternative to the standard deghosting process. We have evaluated the method using both synthetic and field data sets and achieved similar results to an industry-standard FK method while achieving high compression performances.

In summary, our research has demonstrated that constrained dictionary learning-based methods are highly effective in enabling key processing steps, such as wavefield separation and deghosting, to be carried out directly in the compressed domain. While sparse transforms have previously been utilized for some seismic processing steps, there has been no prior proposal for methods aimed at compressing and simultaneously processing seismic data in the compressed domain. Thus, we have shown that such methods can significantly reduce costs related to data storage and transfer, and bring computational cost reduction. Future research can now focus on allowing other key processing steps in the compressed domain.

Sammendrag

Seismiske leteundersøkelser dekker stadig større områder og benytter et økende antall sensorer for å samle inn data. Dette betyr igjen at de seismiske datasettene øker raskt i størrelse. Denne økningen i datavolum utgjør imidlertid en betydelig utfordring for konvensjonelle seismiske prosesserings- og bildeteknikker, som krever omfattende menneskelige og beregningsmessige ressurser. Komprimering av de seismiske dataene på et tidlig stadium av den seismiske prosesseringssekvensen kan være et nøkkelelement for å overvinne lagrings- og dataoverføringsbarrierer. Videre vil bruk av seismiske prosesseringstrinn direkte i det komprimerte domenet ikke bare spare lagrings- og overføringskostnader, men kan også føre til raskere og mer kostnadseffektive alternativer til standard seismisk prosessering.

Den relevante informasjonen i et seismiske datasett er av mindre dimensjonalitet enn dataene selv. Følgelig kan seismiske data representeres ved hjelp av et redusert antall koeffisienter sammenlignet med det totale antall datapunkter ved å transformere dataene til et passende matematisk domene. Generelt kalles slike transformasjoner glisne ('sparse') og de er blitt gjenstand for en større interesse innen seismisk prosessering i løpet av de siste årene. Denne oppgaven undersøker først ulike glisne representasjoner for anvendelser innen seismisk datakomprimering, hvor det skilles mellom fast definerte og innlærte transformasjoner. Deretter diskuteres ulike transformasjoner som er basert på analytiske beskrivelser av det seismiske bølgefeltet . Fokus vil spesielt være på en parabolsk beskrivelse av de seismiske bølgene ved hjelp av lokale kinematiske parametre. De ekstraherte kinematiske parametrene benyttes til å utvikle nye prosesseringsmetoder basert på enkle operatorer som anvendes direkte på de komprimerte seismiske data.

I denne oppgaven er følgende hovedresultater blitt oppnådd: (1) En 'Dictionary Learning' (DL)-basert komprimeringsmetode er utviklet, hvor redundans i seismiske data utnyttes fullt ut ved å lære detaljerte egenskaper ut fra lokale tidsvinduer av seismiske data. Metoden har blitt testet ved bruk av både syntetiske data og feltdata og er vist å være mer effektiv sammenlignet med konvensjonelle komprimeringsmetoder, som er basert på forskjellige forhåndsdefinerte transformasjoner. (2) En ny metode for dual-sensor bølgefeltseparasjon i det komprimerte domenet er utviklet for 2D data basert på parabolsk DL. Kinematiske parametere som helning og krumning til DL-atomene gjør det mulig å utføre bølgefeltseparasjon direkte i det komprimerte domenet. Metoden gir sammenlignbare resultater med en industristandard FK-basert metode for bølgefeltseparasjon og har fordelen av å være robust mot romlig aliasing uten behov for interpolering og gir samtidig høy kompresjonsytelse. (3) En dehosting-prosess i det komprimerte domenet, spesielt designet for lave frekvenser, er også utviklet. Metoden kan sees på som komplementær til bølgefeltseparasjonsmetoden som vanligvis kun håndterer høye frekvenser. Ved

å operere direkte på det komprimerte dataformatet, som er av mindre størrelse, representerer denne nye metoden et kostnadseffektivt alternativ til en standard deghosting-prosess. Vi har evaluert metoden ved å benytte både syntetiske data og feltdata og har oppnådd sammenlignbare resultater med en industristandard FK-metode samtidig med en høy kompresjonsytelse.

Kort oppsummert har forskningen vår vist at læringsbaserte DL-metoder er svært effektive i forhold til å utføre viktige prosesseringstrinn, som bølgefeltseparasjon og deghosting, direkte i det komprimerte domenet. Mens glisne transformasjoner tidligere har vært benyttet i noen seismiske prosesseringstrinn, har det ikke tidligere blitt foreslått metoder skreddersydd for å komprimere og samtidig prosessere seismiske data i det komprimerte domenet. Vi har demonstrert at denne type metoder kan redusere kostnadene knyttet til datalagring og overføring betydelig, og gi en beregningsmessig kostnadsreduksjon. Videre forskning kan nå fokusere på å utvikle andre viktige seismiske prosesseringssekvenser i det komprimerte domenet.

List of Publications

Paper I

Faouzi Zizi, M. O., and P. Turquais. "A dictionary learning method for seismic data compression". Published In: *Geophysics*. Vol. 87, no. 2 (2022), pp. 101-116. DOI: 10.1190/GEO2020-0948.1.

Paper II

Faouzi Zizi, M.O., P. Turquais, A. Day, M. W. Pedersen, and L. J. Gelius. "Dual-sensor wavefield separation in a compressed domain using parabolic dictionary learning". Published In: *Geophysical Prospecting*. (2023), pp. 1-19. DOI: 10.1111/1365-2478.13348.

Paper III

Faouzi Zizi, M.O., P. Turquais, A. Day, M. W. Pedersen, and L. J. Gelius. "Low frequency seismic deghosting in a compressed domain using parabolic dictionary learning". *Submitted to Geophysical Prospecting for publication.*

Conference papers

1. Faouzi Zizi, M. O., and P. Turquais, 2020, A dictionary learning method for seismic data compression: *2nd EAGE Marine Acquisition Workshop, Oslo, Norway.*

2. Faouzi Zizi, M.O., P. Turquais, A. Day, M. W. Pedersen, and L. J. Gelius, 2022, Parabolic dictionary learning for seismic dual-sensor wavefield separation in a compressed domain: *Lofotseminaret 2021 in Applied Geophysics, Nyvågar, Norway.*

3. Faouzi Zizi, M.O., P. Turquais, A. Day, M. W. Pedersen, and L. J. Gelius, 2022, Simultaneous dual sensor wavefield separation and seismic data compression using parabolic dictionary learning: *83rd EAGE Annual Conference and Exhibition, Madrid, Spain.*

Contents

Chapter 1

Introduction

1.1. Motivation

In the marine seismic industry, vast amounts of data are collected from tens of thousands of sensors, resulting in data sets that often reach several terabytes in size. This presents potential bottlenecks during data transfer and processing. For example, during processing the data are duplicated several times to allow proper quality control (QC) and enable repeating processing steps, which requires additional disk space and management of data. Additionally, the large data size makes direct data transfer between onshore processing centers time-consuming, and between vessels and processing centers almost impossible. Indeed, the satellites available bandwidth is not enough to enable the transfer of all pre-stack data recorded on the vessels. Hence, physical tapes are still the most commonly used way for transferring data, either upon completion of the acquisition process or during crew changes. An efficient compression technique

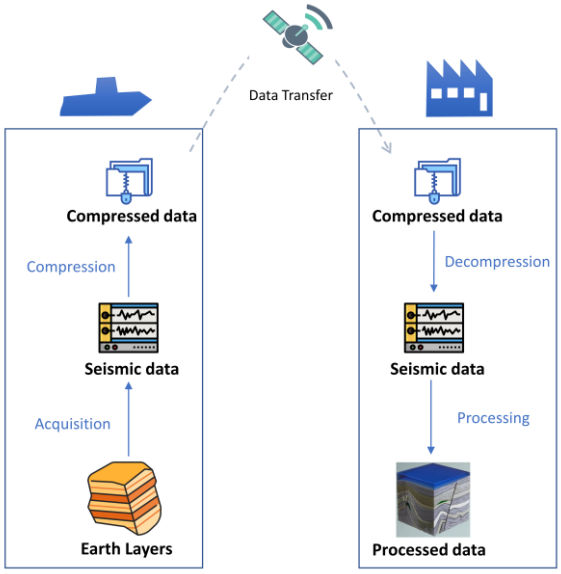

Figure 1.1: *Schematics of the conventional use of seismic data compression in the context of marine seismic exploration.*

would save costs associated with disk storage and data transfer, thereby removing barriers to faster product delivery. For instance, efficient compression could: enable data transfer from the vessel to an onshore processing center before the end of acquisition; reduce the size of long-term data storage on disk, particularly for data that remains unused for an extended period.

Figure 1.1 shows how a seismic data compression algorithm can be used in the context of marine seismic acquisition and processing. Although such algorithms can help to achieve various goals, it is important to note that data decompression is always necessary before applying any processing step. This is because conventional processing steps typically work with input data in their original time-space domain. As a result, the benefits of compressing seismic data are mainly limited to data transfer and long-term storage. Hence, even if the data is compressed, the cost-saving benefits related to disk storage or project turnaround time are not realized during processing. Enabling steps of the processing sequence directly in a compressed domain can overcome the requirement for data decompression and thus not only save costs related to storage during processing, but also provide a faster and cheaper alternative to the standard seismic processing sequence since the processing would be applied on compressed data which are of smaller size. Figure 1.2 shows how the general workflow would look like if processing steps were enabled in the compressed domain.

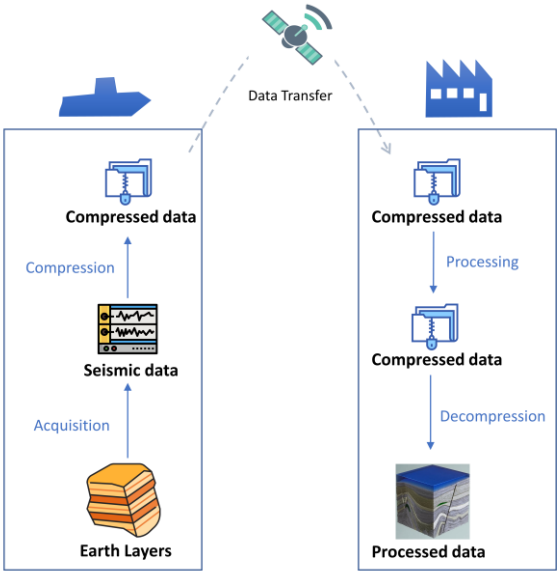

Figure 1.2: *Schematics of the use of a seismic compression method where processing steps can be carried out directly in the compressed domain in the context of marine seismic exploration.*

1.2. Objectives and outline of the thesis

1.2.1. Thesis objectives

Data compression algorithms are either lossy or lossless. Common lossless data compression algorithms like "ZIP" (Deutsch, 1996) recover an identical copy of the input after compression and decompression, while lossy algorithms like "JPEG" (Taubman and Marcellin, 2002) result in lower reconstructed data quality. As the marine seismic industry requires high compression rates to reach the aforementioned goals, we focus on lossy data compression algorithms in this thesis. Several data compression algorithms have been developed to compress seismic data. Those algorithms are generally based on mathematical fixed transforms/domains such as Fourier, Wavelets, Curvelets, Discrete Cosine, and others. Dictionary Learning (DL) methods are another type of transforms not based on mathematical fixed domains. While some of these algorithms have been recently used in the marine seismic industry for various tasks such as noise suppression (Beckouche and Ma, 2014; Turquais et al., 2017a; Turquais et al., 2017b) or interpolation (Turquais et al., 2018), they are not specifically tailored for seismic data compression. Yet, such methods have great potential for data compression considering the high level of sparsity they provide (Skretting and Engan, 2011; Akbari and Trocan, 2019).

While seismic data compression is a major concern, it is important not to overlook the impact of seismic processing on disk storage and project turnaround time. Seismic processing refers to a series of geophysical techniques used to eliminate unwanted signals from recorded data and to relocate the seismic energy to the point where it was originally diffracted or reflected from. Figure 1.3 illustrates the main steps of a simple processing sequence after acquisition. Most of the seismic processing methods are carried out in a different domain than the time-space domain. Transforming the data into another domain often requires preconditioning (e.g., data interpolation, zero padding), which comes at a significant computational cost. Moreover, throughout the long processing time,

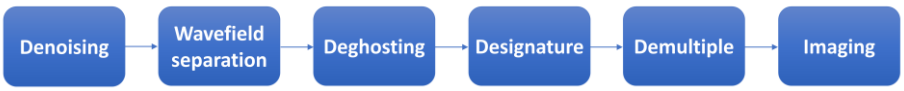

Figure 1.3: *A simple processing workflow after acquisition including imaging. First data are denoised and separated into up- and down-going parts of the wavefield. Then the deghosting, designature, and demultiple processes remove the effects of the source signature and the sea surface. Finally, the data are migrated to obtain an image of the subsurface.*

data is duplicated numerous times and maintained in an uncompressed format on disk. Furthermore, after each processing step, the data are transformed back into the original time-space domain for quality control. Hence, seismic processing and imaging results in a high quality multidimensional image of the subsurface but is expensive in terms of human and computational resources. Processing seismic data directly in a compressed domain has the potential to save disk storage, human costs, and computational resources. Thus, the first step of this thesis is to (1) develop a compression algorithm specifically designed for seismic data based on a transform providing a high level of sparsity. Such algorithm is designed with the aim of reaching high compression performances while maintaining high seismic data quality and should fit into the illustration shown in Figure 1.1, i.e., not with the aim of processing data in the compressed domain. Then, we (2) identify a sparse transform which enables analysis of the kinematics of the seismic events directly in the sparse domain and thereby build a bridge between data compression and seismic processing. Such transform will be used to (3) develop novel methods that apply the wavefield separation and receiver deghosting processing steps directly in the compressed domain. These steps were chosen over other steps (see Figure 1.3) because they naturally fall near the beginning of a standard marine seismic processing sequence. Yet, the denoising process was not brought into focus in this thesis because this step has been extensively studied in the literature by means of sparse transforms (Hennenfent and Herrmann, 2006; Neelamani et al., 2008; Beckouche and Ma, 2014; Turquais et al., 2017a; Turquais et al., 2017b).

1.2.2. Outline

Chapter 2 of this thesis focuses on sparsity-promoting transforms, both learned and fixed, that can be used for seismic data compression. The chapter begins by discussing the exact and approximate sparse problems and how they relate to compression. It then describes various efficient compression techniques, known as entropy coding, which can be used in conjunction with sparsity-promoting transforms to achieve effective compression of seismic data. Chapter 3 proceeds by exploring various transforms that can accurately describe the seismic wavefield, based on their analytical expressions. Particular emphasis is placed on a constrained version of dictionary learning methods known as parabolic dictionary learning. This approach focuses on locally describing the seismic wavefield through kinematic parameters. Chapter 4 starts by presenting conventional methods for wavefield separation and receiver deghosting, which are the processing steps we want to enable in the compressed domain in this thesis. Then, it proposes simple operators that can be applied in the sparse domain to enable these processing steps in the compressed domain. Chapter 5

summarizes the main scientific contribution of the three papers written during this thesis. Finally, Chapter 6 presents some discussions, conclusions, and an outlook.

Chapter 2

Seismic data compression

Seismic data compression algorithms generally involve a three-stage process, which includes transformation, quantization, and coding. The latter two stages are mathematical techniques generally referred to as entropy coding. The decompression algorithms perform the inverse process of each stage, leading to the reconstruction of the original data. Figure 2.1 depicts the different stages of seismic data compression, including the data reconstruction process. In this study, we build upon these fundamental principles to address the seismic compression problem.

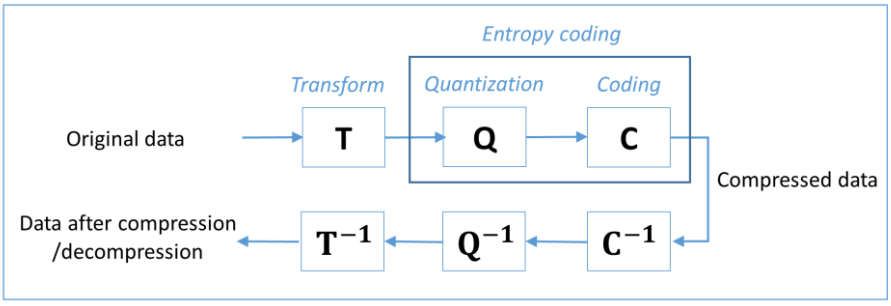

Figure 2.1: *Main stages of a compression / decompression algorithm. Here the superscript '-1' refers to the inverse operations. Note that the inverse quantization denoted* Q^{-1} *does not include reversing rounding errors.*

2.1. Transformation to a sparse domain

2.1.1. The exact sparse representation problem

The transformation stage allows the representation of seismic data in another mathematical domain. In order to provide data compression, this representation in the transformed domain must be more compact (including fewer large amplitude coefficients and more small amplitude coefficients) than the original data. A synthesis sparse representation of the data is composed of two parts: a dictionary and a sparse coefficient vector. The dictionary is a collection of vectors called atoms representing patterns or waveforms. The sparse coefficient vector

stores a relatively small number of non-zero coefficients that contains the information related to the amplitude and location of the waveforms in the data. Multiplying the dictionary and the sparse coefficient vector associates the morphological waveforms to their location and amplitude and thereby reconstructs the data. The analysis approach also exists, where data are multiplied by dictionary atoms to provide a sparse representation. Details regarding the differences between both approaches can be found in Elad et al. (2007). In this thesis we consider only the synthesis approach. In order to understand how the data are represented in a sparse domain, we consider a simple 1D example as illustrated in Figure 2.2. Here, the original 1D signal of size N is denoted $\mathbf{y}$. The transform domain is defined by a dictionary of atoms $\mathbf{D} = [\mathbf{d_1}, \mathbf{d_2}, ..., \mathbf{d_k}]$ of the same size N as the original signal $\mathbf{y}$, and a sparse coefficient vector denoted $\mathbf{x}$ of size K. The exact sparse representation problem consists of finding the vector $\mathbf{x}$ with the least number of non-zero coefficients such that multiplying $\mathbf{D}$ by $\mathbf{x}$ results in a linear combination of atoms that reconstructs the original signal $\mathbf{y}$. Hence, the exact sparse representation problem can be mathematically expressed as:

$$\min_{\mathbf{x}} \| \mathbf{x} \|_0 \text{ subject to } \mathbf{y} = \mathbf{D}\mathbf{x} \tag{2.1}$$

In this equation, the ℓ_0-norm is not a proper mathematical norm; it has been redefined as the number of nonzero coefficients in a vector or an array (Donoho and Elad, 2003; Donoho, 2006). For example, Figure 2.2 shows that the signal $\mathbf{y}$ can be represented as a linear combination of the Discrete Cosine Transform (DCT) dictionary atoms. Indeed, multiplying $\mathbf{d_4}$ by -1 and $\mathbf{d_8}$ by 0.5, followed by a summation of both vectors reconstructs exactly the original signal $\mathbf{y}$.

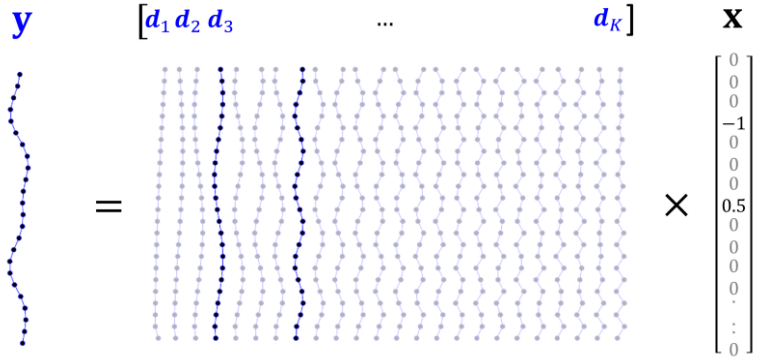

Figure 2.2: *Illustration of a simple one dimensional signal reconstruction by solving the exact sparse representation problem.*

2.1.2. The sparse approximation problem

In general, it is challenging to reconstruct the geophysical signal with a strictly sparse representation, i.e., a representation where a significant amount of amplitude coefficients are exactly equal to zero. Consequently, striving for an exact solution leads to a signal representation that is not sparse in the strict sense, implying that the solution contains only a few coefficients exactly equal to zero. Figure 2.3 shows the reconstruction of a signal $\mathbf{y}$ (in green) which simulates a geophysical signal with a more complex structure than that shown in Figure 2.2. The solution depicted represents the signal precisely, but, unfortunately, is not very sparse since $\mathbf{x}$ has only a few zero coefficients. Consequently, it is often preferred to tolerate a small representation error, with the aim of finding a representation which is sparser. For example, in Figure 2.4 only two coefficients with the highest values were kept in order to approximately reconstruct the signal $\mathbf{y}$. This is referred as the sparse approximation problem. In order to address this problem, we can look for the sparsest solution while tolerating an error below a fixed threshold ε (Bruckstein et al., 2009). This approach will be referred as the "*error constraint mode*" and can be mathematically expressed as:

$$\min_{\mathbf{x}} \| \mathbf{x} \|_0 \text{ subject to } \quad \| \mathbf{y} - \mathbf{D}\mathbf{x} \|_2 \leq \varepsilon \tag{2.2}$$

An alternative solution is to restrict the number of non-zero coefficients in the vector $\mathbf{x}$. This restriction sets a maximum number of non-zero coefficients T for the atoms and aims to find a solution that minimizes the error (Tropp, 2004). Such an approach will be referred as the "*sparsity constraint mode*" and can be mathematically expressed as:

$$\min_{\mathbf{x}} \| \mathbf{y} - \mathbf{D}\mathbf{x} \|_2 \text{ subject to } \quad \| \mathbf{x} \|_0 \leq T \tag{2.3}$$

When the dictionary is an orthonormal basis, the exact and sparse approximation problems can be solved by hard thresholding (Elad, 2010). If $N < K$, the dictionary is said to be redundant. The greater the redundancy in a dictionary, the sparser and more accurate will be the representations (Donoho, 2006). However, in this case the solution will not be unique. This problem is known to be NP-hard and can be solved with matching pursuit algorithms such as the orthogonal matching pursuit (OMP) (Pati et al., 1993), which is used in this thesis.

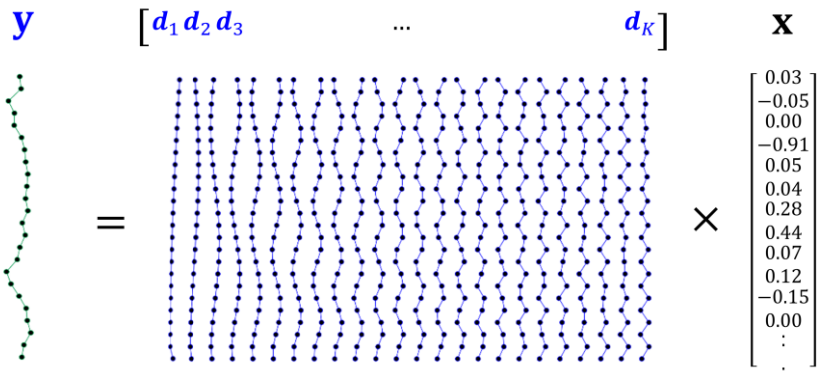

Figure 2.3: *Illustration of a realistic one dimensional signal reconstruction by solving the exact sparse representation problem.*

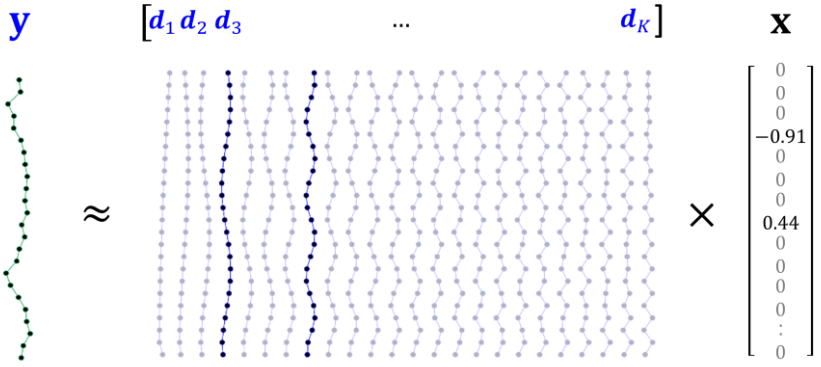

Figure 2.4: *Illustration of a realistic one dimensional signal reconstruction by solving the sparse approximation problem.*

2.2. Dictionaries for sparse representations

2.2.1. Predefined dictionaries

In order for data compression to be effective, a high degree of sparsity must be present in the transformed domain. To obtain a sparse representation of the data, conventional methods involve selecting pre-defined or fixed atoms from the basis vectors of an analytically defined domain. Pre-defined dictionaries such as Wavelets, Curvelets, and Discrete Cosines (Mallat, 2008) are commonly used to develop compression algorithms (Elad, 2010). For example, the JPEG compression format is based on the Discrete Cosine Transform (DCT). These pre-defined dictionaries have also been used for the purpose of compressing seismic data (Duval and Nguyen, 1999; Duval and Rosten, 2000; Wang et al., 2004; Zheng and Liu, 2012; Fajardo et al., 2015). In this section we investigate two main pre-defined dictionaries, the Discrete Cosine Transform (DCT), and the Wavelet Packet Transform (WPT). These two transforms are used in two seismic data compression algorithms that were used for benchmarking, presented in Paper I.

The DCT representation expresses a signal as a combination of sinusoidal waveforms with varying amplitudes and frequencies, thereby transforming the signal from its spatial domain to its frequency domain. The basis functions of the Discrete Cosine Transform (DCT) are essentially the real part of the Fourier basis functions, as they use only real-valued cosine functions. In mathematical terms, the basis functions of the 2D DCT transform are defined as follows:

$$\mathbf{Y} = \sum_{n1=0}^{\mathrm{N1}-1} \left(\sum_{n2=0}^{\mathrm{N2}-1} \mathbf{y}_{n1,n2} \cos\left[\frac{\pi}{\mathrm{N2}} \left(n2 + \frac{1}{2} \right) k2 \right] \right) \cos\left[\frac{\pi}{\mathrm{N1}} \left(n1 + \frac{1}{2} \right) k1 \right] \qquad (2.4)$$

where $\mathbf{Y}$ is the signal representation in the transform domain, $\mathbf{y}$ is the original signal representation in the spatial domain, $N1 \times N2$ is the size of the original 2D signal, $k1 \times k2$ is the size of the 2D signal in the transform domain. Compression methods based on the DCT rely on the principle that the low-frequency components of a signal are more significant than the high-frequency components (W. Smith, 1997). Hence, reducing the number of bits required to represent the high-frequency component can cause only a slight degradation in the image quality. Figure 2.5-a illustrates an 8 by 8 2D DCT.

The WPT (Fajardo et al., 2015) is a type of wavelet transform that involves passing a signal through low and high-frequency filters. When applying this transform to an input image, a forward transform can be achieved through a series of cascaded filtering steps. The image is first filtered by a set of complementary filters (i.e., a low-pass filter and its complementary high-pass

filter), which preserves the entire frequency content of the image when the filtered components are combined. The low-pass filtered image is then filtered by another set of complementary filters, as is the high-pass filtered image, and so on. The Haar wavelet packet transform (Elad, 2010) is obtained by filtering an image with simple 1D filters along the horizontal and vertical axes. Figure 2.5-b depicts the filters used in an 8 by 8 2D Haar Transform. For a more detailed mathematical explanation of the wavelet transform, refer to Sweldens, (1995) or Mallat (2008).

2.2.2. Dictionary learning

Transforming seismic data to a sparse domain based on pre-defined dictionaries may not be ideal since the basis vectors of a selected fixed transform are not designed with the aim of representing the seismic signal. Although, traditional transforms may provide satisfactory sparsity performance, they may not be optimal when used with seismic waveforms related to complex structures. Dictionary learning (DL) methods, e.g., K-mean Singular Value Decomposition (K-SVD) (Aharon et al., 2006) or Online Dictionary Learning (ODL) (Mairal et al., 2009), provide other types of dictionary that are not based on mathematical fixed transforms/domains. These methods capture redundant events from the original data, store them in a dictionary of waveforms, and then express the signal as a weighted sum of a few of the learned waveforms. Hence, such methods have the potential to fully exploit the high level of similarity in seismic data providing representations with a higher level of sparsity than conventional dictionaries based on fixed transforms. Consequently, DL approaches are expected to be more effective for seismic data compression. Figure 2.5-c shows an example of an 8 by 8 2D dictionary learned from seismic data.

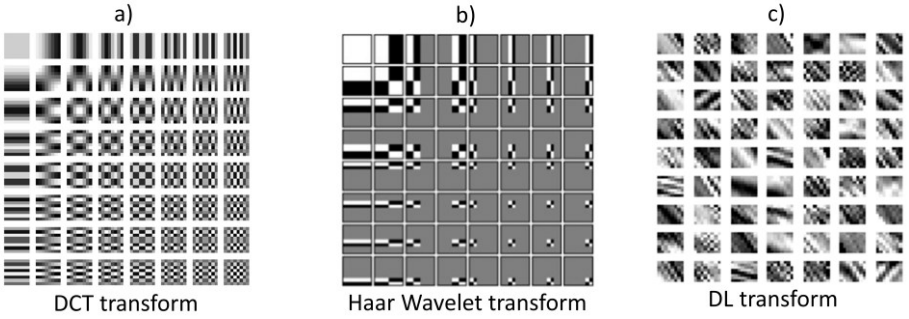

Figure 2.5: *Illustration of 8 by 8 2D dictionaries. a) the Discrete Cosine Transform, b) the Haar Wavelet Transform, c) a Dictionary Learning Transform.*

In DL problems, the objective is to represent the original data in a sparse fashion using two key components: a dictionary of learned atoms that represent elementary waveforms that occur repeatedly in the data, and a set of sparse coefficient vectors. Let us consider a data set of L signals denoted $\{\mathbf{y_i}\}_{i=1}^{L}$ each of size N. The first step is to extract a training set denoted $\{\mathbf{y_i}\}_{i=1}^{M}$ which is a sub-set of the original data set comprising M signals, where $M < L$. Then, apply a DL method to jointly: learn a redundant dictionary $\mathbf{D} \in \mathbb{R}^{N \times K}$ where $N < K$ (Donoho and Elad, 2003) of K atoms, each of size N, with $K < M$; and find the set of sparse coefficient vectors $\mathbf{x}_1, \mathbf{x}_2, .. \mathbf{x}_M$ that minimize the representation error given an error constraint ε or a sparsity constraint T imposed on the sparse coefficient vectors (Aharon et al., 2006). When the 'error constraint mode' is applied, this problem can be mathematically expressed by

$$\min_{\{\mathbf{x_i}\}_{i=1}^{M},D} \sum_{i=1}^{M} \| \mathbf{x_i} \|_0 \text{ subject to } \quad \| \mathbf{y_i} - \mathbf{D}\mathbf{x_i} \|_2 \leq \varepsilon, \qquad i = 1, ... M \ . \tag{2.5}$$

When the 'sparsity constraint mode' is applied, this problem can be mathematically expressed by

$$\min_{\{\mathbf{x_i}\}_{i=1}^{M},D} \sum_{i=1}^{M} \| \mathbf{y_i} - \mathbf{D}\mathbf{x_i} \|_2^2 \text{ subject to } \quad \| \mathbf{x_i} \|_0 \leq T, \qquad i = 1, ... M \ . \tag{2.6}$$

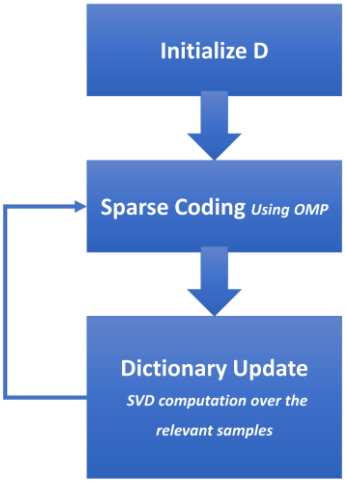

Figure 2.6: *Illustration of the k- mean singular value decomposition (KSVD) algorithm main steps.*

Figure 2.6 illustrates the main steps of the K-SVD algorithm, which involves initializing the dictionary by selecting random atoms from the extracted patches of the initial data set. Next, the Orthogonal Matching Pursuit (OMP) algorithm is applied to solve the sparse problem (equations 2.2 or 2.3). The dictionary atoms are then updated one by one using Singular Value Decomposition (SVD). Once the whole dictionary is updated, another OMP is performed with the new dictionary to obtain a sparser representation, and the process is repeated iteratively (indicated by the arrow in Figure 2.6). The iteration between OMP and dictionary updating stops after a fixed number of iterations defined by the user. Once the dictionary $\mathbf{D}$, is learned, OMP is again used to solve the sparse optimization problem, but this time for the L signals in the original data set $\{\mathbf{y_i}\}_{\mathrm{i=1}}^{L}$. This procedure allows us to obtain sparse coefficient vectors $\mathbf{x}_1, \mathbf{x}_2, .. \mathbf{x}_L$, which can be used to represent the original data in the sparse DL domain.

2.3. Entropy Coding

After the signal has been transformed to a sparse domain, the entropy coding stage follows, which generally includes two tasks: quantization and coding. Quantization is a lossy compression strategy that is used to achieve compression with an approximation of the floating-point coefficients (the non-zero sparse coefficient for the presented method) by a set of integers. This means that the amplitude scale is discretized during quantization. The quantization process involves two operations: rescaling and rounding. During rescaling, the sparse coefficients are multiplied by a scaling coefficient (Sc), and then the coefficients are rounded to enable them to be coded with fewer bits. The quantization can be uniform or non-uniform (Fajardo et al., 2015). Figure 2.7 provides an example of both quantizations. In this example, [a1, a2, a3, ..., ak] represents sparse coefficients before quantization, and [b1, b2, b3, ..., bk] represents the quantized values of these coefficients. For instance, the amplitude coefficients in a range [a1, a2] are discretized into the same value b2. Uniform quantization may be more appropriate for seismic compression because the larger errors in non-uniform quantization occur in the larger amplitude coefficients, which are likely to contain relevant geophysical information in seismic data. By contrast, the small amplitude coefficients may correspond to noise in the seismic data.

After quantization, a coding scheme can be applied to the quantized coefficients in order to seek a data representation with a small number of bits. In the field of data compression, Huffman and Arithmetic Coding are the most commonly used coding schemes, with the goal of forming symbol sequences in a way that allows them to be coded with fewer bits. Among these, the Huffman

coding scheme (Huffman, 1952) is considered one of the most popular and efficient. This scheme creates variable-length codes, with each code represented by an integer number of bits. Symbols with higher probabilities are assigned shorter code words, and the scheme is particularly effective when quantized amplitude coefficients occur with different probabilities (Skretting, et. al, 1999). Table 1 shows a simple example in which some characters occur more frequently than others in a set of data represented by letters. When Huffman coding is used, letters with high frequencies (such as 'B', 'F', and 'C') are coded with fewer bits (e.g., '01', '00', and '10') compared to other letters with low frequencies (e.g., 'D' and 'G'), which are coded with '11111' and '11110'. However, when a standard coding scheme based on fixed length is used, the frequencies of occurrence are not considered. After calculating the total number of bits (by multiplying the length of the codes with their corresponding frequencies) based on the simple example depicted in Figure 1, we find out that the Huffman scheme indeed codes these data using a smaller number of bits (146 instead of 174).

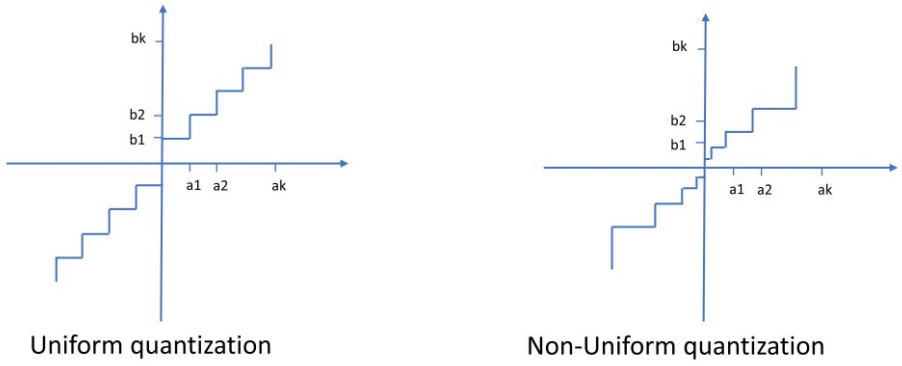

Figure 2.7: *Illustration of uniform and non-uniform quantization (adapted from Fajardo et al., (2015)).*

Data	A	B	C	D	E	F	G	**Total number of bits**
Frequency	10	15	12	3	4	13	1	
Code with fixed length	‘000’	‘001’	‘010’	‘011’	‘100’	‘101’	‘110’	174
Code with Huffman	‘110’	‘10’	‘00’	‘11111’	‘1110’	‘01’	‘11110’	146

Table 1: *Comparison between fixed length coding and Huffman coding strategies based on a simple data set where characters occur with different frequencies.*

Chapter 3

Transforms for seismic processing

3.1. Predefined vs learned dictionaries

In the previous chapter, we presented some predefined dictionaries whose basis vectors are based on fixed mathematical transforms, such as the Discrete Cosine (DCT) and Haar wavelet transforms. These transforms were used to compute sparse representations of seismic data. Other transforms, such as curvelets (Candes and Donoho, 2000, 2002) or seislets (Fomel and Liu, 2010) have also been used to reconstruct seismic data via sparsity-promotion given their high potential to describe the seismic signal morphology. Indeed, atoms of a dictionary should describe the morphology of a seismic signal in order to provide a representation with a high level of sparsity. The more closely the morphology of the atoms reflects the morphology of the seismic waveforms, the higher the level of sparsity. For example, Fourier bases are frequently chosen to describe the seismic wavefield (Zwartjes and Sacchi, 2007; Schonewille et al., 2009; Naghizadeh and Sacchi, 2010a) mainly because the basis vectors represent monochromatic waves, and a monochromatic plane wave is the simplest solution to the wave equation in a homogenous medium (Kiselev, 2007). Wavelet transforms have also been used for many seismic data applications (Mallat, 2008), such as compression and denoising (Villasenor et al., 1996; Fajardo et al., 2015; Mohanalin et al., 2016). Their basis vectors are known to be well localized in time and thus provide a superior representation of transient signals than the Fourier basis for some seismic processing tasks (Forster et al., 1994). Curvelets are an extension of wavelets, which are scaled, localized and directional at fine scales. Hence, they are able to better represent seismic patterns which are curved compared to wavelets (Candes and Demanet, 2005). Additionally, curvelets have been shown to be an effective basis to describe, denoise, and interpolate seismic data (Hennenfent and Herrmann, 2006; Naghizadeh and Sacchi, 2010b). These bases have also been used for seismic imaging (Douma and De Hoop, 2007; Chauris and Nguyen, 2008).

In the preceding chapter, we have also explored Dictionary learning (DL) methods, which have the potential to represent seismic data with higher sparsity levels than conventional dictionaries based on fixed transforms. DL methods learn seismic waveforms directly from seismic data, thus allowing for a better

description of the seismic signal morphology (Elad, 2010). Consequently, DL methods have already been used for seismic processing tasks such as denoising (Beckouche and Ma, 2014; Turquais et al., 2017). While sparsity is a vital component for accomplishing many seismic processing tasks, it is not always sufficient. Although conventional DL methods have the potential to generate highly sparse representations, the absence of an analytic expression for the learned atoms limits the range of processing steps that can be performed in the DL domain. Interpolation to arbitrary locations is an example of a processing step that requires an analytic expression. Fourier or Curvelets transforms, which have basis functions with analytic expressions, can be used to interpolate seismic data (Xu et al., 2005; Naghizadeh and Sacchi, 2010b). That is because analytic functions can be evaluated at any point. In contrast, learned atoms are only defined at discrete positions, and the representation of data as a linear combination of atoms is constrained to these same discrete positions. Therefore, the lack of an analytic expression makes it impossible to interpolate data over an arbitrary grid. Thus, to accommodate most processing steps, DL methods would require atoms with analytic expressions.

3.2. Physics-constrained dictionary learning

With the goal of combining the benefits of predefined dictionaries, which define atoms using analytical expressions, with the high sparsity levels achieved through DL methods, Turquais et al. (2018) proposed a novel approach. The authors proposed to constrain the morphology of dictionary-learned atoms to have a parabolic shape, thereby creating a parabolic dictionary learning (PDL) method that could leverage the strengths of both predefined and learned atoms. The parabolic structure was chosen because it is consistent with the physics of the wavefield propagation. Previous studies by Bortfeld (1989), Hubral et al. (1992), Cerveny (2001), and Zhang et al. (2001) have shown that a seismic wavefield recorded at the surface can be described as a sum of waves whose travel times are analytically represented by parabolic or hyperbolic moveouts. Indeed, according to the ray theory (Ursin, 1982; Hubral, 1983), if a central ray travels from a source located at x_s to a receiver located at x_r, while an arbitrary paraxial ray travels in the vicinity of the central ray from a source located at $x_s{}'$ to a receiver located at $x_r{}'$, and under the assumption of smoothly changing amplitudes in the earth model, the travel time of that paraxial ray can be expressed as:

$$t(\Delta x_s, \Delta x_r) = t_0 + s_r \Delta x_r - s_s \Delta x_s + \tfrac{1}{2} B^{-1} A \Delta x_s{}^2 + \tfrac{1}{2} D B^{-1} \Delta x_r{}^2 - B^{-1} \Delta x_s \Delta x_r \quad (3.1)$$

where $\Delta x_s = x'_s - x_s$, $\Delta x_r = x'_r - x_r$, s_s and s_r are the slowness of the paraxial ray at the source and the receiver, respectively. A, B and D are scalar matrix elements of the 2 by 2 ray propagator matrix (Cerveny, 2001). Equation (3.1) is known as the parabolic travel time formula and represents a second order approximation of the ray travel time. It was introduced by Bortfeld (1989) for a 2D medium, then generalized by Hubral et al., (1992) to 3D laterally inhomogeneous media. In the common shot configuration where $\Delta x_s = 0$, equation (3.1) can be rewritten as follows:

$$t(\Delta x_s = 0, \Delta x_r) = t_0 + s_r \Delta x_r + \tfrac{1}{2} D B^{-1} {\Delta x_r}^2. \tag{3.2}$$

The formula in (3.2) describes the equation of a parabola. Therefore, based on this description of the wavefield, Turquais et al. (2018) learned dictionary atoms in the common shot domain, and constrained them to have a parabolic shape. This constraint enabled the interpolation of 3D streamer data in the crossline direction, leveraging the strengths of both predefined and learned atoms for more efficient and accurate seismic interpolation results.

In order to learn parabolic atoms, the authors have added an extra constraint during the dictionary update stage where atoms are updated one by one using singular value decomposition (Figure 2.6). When an atom is learned, and before updating it, PDL finds the parabolic parameters which maximize the trace-to-trace correlation of the atom. Consider an atom representing a small number of traces as shown in Figure 3.1, and consider the blue parabola in the figure to have the following travel time Δt of parabolic moveout:

$$\Delta t = b\ \Delta x_r + c\ {\Delta x_r}^2 \tag{3.3}$$

$\Delta x_r = x_r^i - x_r^{ref}$ is the spatial shift between the receiver located at a trace i and the receiver located at the center or reference trace, b is the slope of the parabolic function, and c the curvature of the parabolic function. The fitting is done by finding the parabolic parameters b and c that minimize the error $\left\| \Delta t - b\ \Delta x_r - c\ {\Delta x_r}^2 \right\|_2$. Hence, after fitting, each atom is characterized by the parabolic parameters b and c. Then, PDL averages the samples of the atom that lie on the same parabola by flattening the event, stacking the traces, and shifting them back according to the parabolic time moveout described in equation (3.3). The end result is atoms with parabolic shape and constant amplitudes along the parabolic moveouts. Consequently, it is possible to define learned atoms on a desired sampling grid and thus enable interpolation as illustrated in Figure 3.1. More details regarding PDL can be found in Turquais et al., (2018).

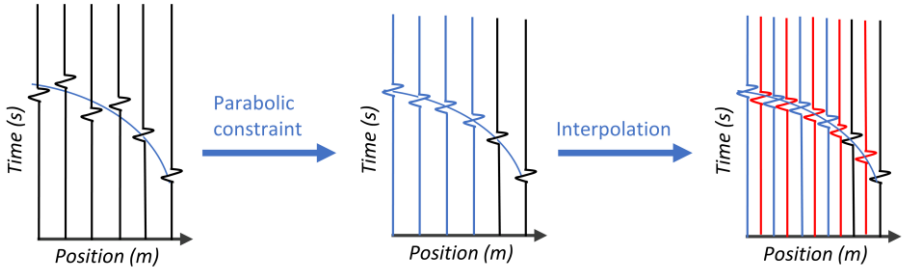

Figure 3.1: *Illustration of an atom representing a small number of traces before and after applying PDL. The figure also shows the resampling of the atom over a desired grid (interpolation).*

In PDL atoms are characterized by parameters such as slope and curvature. In fact, by inspection of equations (3.2) and (3.3), we observe that the parameters of a parabolic moveout can be related to kinematic attributes of the wavefield. Indeed, from (3.2) and (3.3), the slope of a given waveform with parabolic moveout b is identical to the slowness of the ray s_r, and the curvature c is related to elements of the ray propagator matrix (Cerveny, 2001). Hence, Zhang et al. (2001) relate the ray propagator matrix elements to kinematic parameters of the wavefield. For the common shot configuration, we can write:

$$b = \ s_r \ = \frac{\sin\theta_r}{v} \qquad\qquad c = DB^{-1} = \frac{\cos^2\theta_r}{v}. \ \kappa^r. \tag{3.4}$$

where v is the propagation velocity in water, θ_r the angle of incidence of the wavefront at the receiver location, and κ^r the curvature of the wavefront at the receiver location. Zhang et al. (2001) relate the hyperbolic travel time moveout to kinematic parameters of the wavefield not only for the common shot configuration (CS) but also for different configurations, namely, common receiver (CR), common mid-point (CMP), and common offset (CO). Therefore, it is noteworthy that the parabolic travel time moveout approximation is also valid for these different configurations, especially since some processing steps require operating in a domain other than the CS. For these configurations, the parabolic travel time moveout can be related to the kinematic parameters as follows:

$$\begin{cases} CR: \quad \Delta t = b\ \Delta x_s + c\ \Delta {x_s}^2; \quad \Delta x_r = 0 \\ b = -s_s = \dfrac{-\sin\theta_s}{v} \\ c = \dfrac{1}{2}\ B^{-1}A = \dfrac{1}{2}\ \dfrac{cos^2\theta_s}{v}.\kappa^S \end{cases} \tag{3.5}$$

$$\begin{cases} CMP: \quad \Delta t = b\ \Delta h + c\ \Delta h^2; \quad \Delta x_m = 0 \\ b = \dfrac{\sin\theta_s\ +\ \sin\theta_r}{v} \\ c = \dfrac{1}{2}\ (B^{-1}A + DB^{-1} + 2B^{-1}) = \dfrac{1}{2}(\dfrac{cos^2\theta_r}{v}.\kappa^r - \dfrac{cos^2\theta_s}{v}.\kappa^S) \end{cases} \tag{3.6}$$

$$\begin{cases} CO: \quad \Delta t = b\ \Delta x_m + c\ {\Delta x_m}^2; \quad \Delta x_h = 0 \\ b = \dfrac{\sin\theta_r - \ \sin\theta_s}{v} \\ c = \dfrac{1}{2}\ (B^{-1}A + DB^{-1} - 2B^{-1}) = \dfrac{1}{2}(\dfrac{cos^2\theta_r}{v}.\kappa^r - \dfrac{cos^2\theta_s}{v}.\kappa^S) \end{cases} \tag{3.7}$$

where $\Delta x_m = \frac{1}{2}(\Delta x_r + \Delta x_s)$ is the spatial shift in midpoint, $\Delta h = \frac{1}{2}(\Delta x_r\text{-}\Delta x_S)$ the spatial shift in half offset, θ_s the angle of emergence of the wavefront at the source location, and κ^s the curvature of the wavefront at the source location.

Consequently, PDL can be used not only to promote sparsity or describe the morphology of the seismic signals by attributing analytical expressions to the parabolic dictionary atoms, but also to characterize the wavefield from a kinematic perspective.

Chapter 4

Wavefield separation and receiver deghosting using kinematic attributes of the wavefield

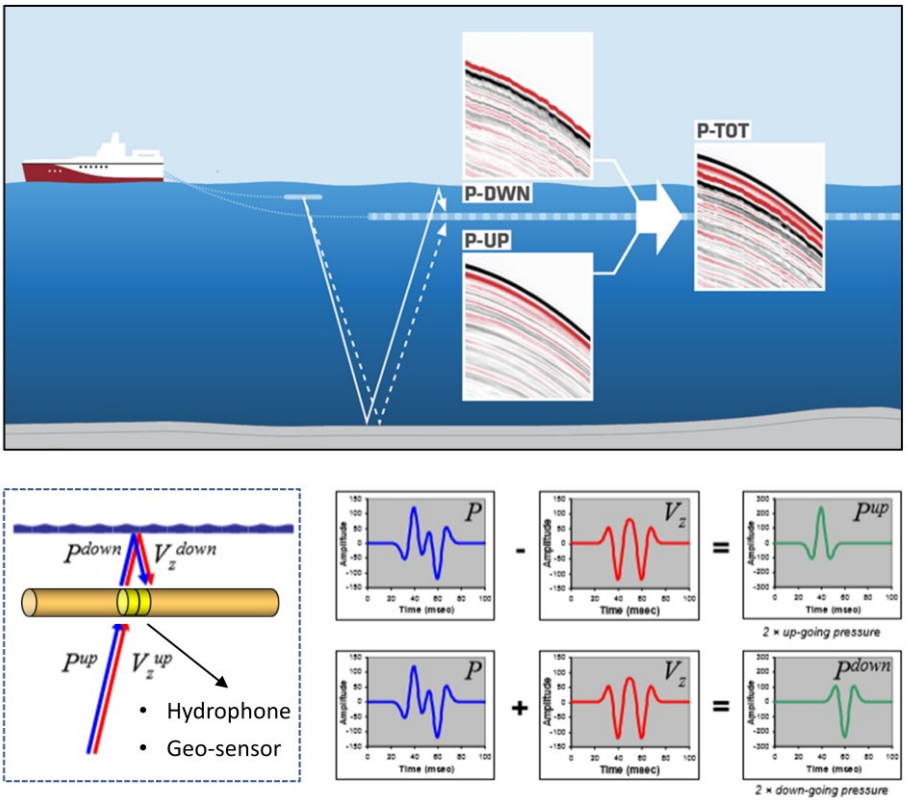

Figure 4.1: *Illustration of primary reflection and its corresponding receiver ghost event as recorded by the dual-sensor acquisition configuration. The figure also demonstrates how the wavefield can be separated into its up- and down-going parts.*

This chapter presents the wavefield separation and receiver deghosting processing steps, emphasizing their importance in the seismic processing sequence. Firstly, we discuss conventional methods that facilitate these processes by determining kinematic attributes such as the incidence angle using plane-wave decomposition. Next, we introduce an alternative approach that defines new operators and applies them directly in a sparse domain characterized by kinematic parameters.

4.1. Conventional methods for receiver ghost removal

Marine seismic data acquired with towed streamers often contain unwanted reflections known as ghosts, which can degrade seismic resolution and reduce the interpretability of subsurface images if not removed from the data. Such reflections arise due to the high impedance contrast between air and water, which causes acoustic waves to reflect at the water surface. These reflections generate both source and receiver ghosts with different periods since sources and streamers are typically deployed at different depths below the water surface (Aytun, 1999; Gosh, 2000). The removal of these unwanted reflections is an essential step in seismic processing to obtain a high-quality subsurface image. In this thesis, we focus only on the receiver-side ghost.

The receiver ghost problem has been a subject of extensive research. Two main categories of methods exist to remove the receiver-side ghost: acquisition-based and processing-based. Conventional single-sensor streamer acquisitions rely on processing-based methods (Amundsen, 1993; Robinson and Treitel, 2008; Masoomzadeh and Woodburn, 2013; Wang et al., 2016) that seek to determine the time delay between the primary and ghost recordings. In contrast, acquisition-based methods, such as dual-sensor streamers (Figure 4.1) (Söllner et al., 2008; Kluver and Day, 2011; Day et al., 2013) and multicomponent streamers (Robertsson et al., 2008; Vassallo et al., 2014), aim to separate the wavefield into up- and down-going parts by combining pressure and particle motion recordings. This process is known as wavefield separation. However, this process alone cannot reconstruct the full data bandwidth, as the low-frequency component of particle motion data is often heavily contaminated with noise. To address this issue, low-frequency particle motion data can be predicted from the relatively clean pressure data, or processing-based methods relying only on pressure measurements can be used.

In the following, we first present the dual-sensor wavefield separation process which is an acquisition-based method. Then, we describe processing-based methods commonly used to remove the receiver-side ghost.

4.1.1. Dual-sensor wavefield separation

Dual-sensor wavefield separation combines the use of hydrophones and geophones or accelerometers. Hydrophones record pressure data while geophones or accelerometers record the vertical component of either particle velocity or acceleration, with the latter being integrated with respect to time to obtain particle velocity. Since pressure is a scalar quantity and the vertical component of particle velocity is a vector quantity, the primary and ghost reflections, which travel in opposite directions, have opposite polarities when recorded with hydrophones, whereas they have the same polarity when recorded with geophones. Figure 4.1 illustrates the acquisition of a primary reflection and its corresponding receiver ghost and depicts their respective waveforms when recorded with hydrophones and geophones. Here, we consider a simple 2D case where the cable is horizontal, and the propagation velocity is constant at the cable depth. By combining both recordings, it is possible to separate the up- and down-going parts of the wavefield. However, before the two recordings can be combined, the amplitudes recorded by the geophone must be scaled by an obliquity factor F, to correct for the fact that only the vertical component of particle motion is recorded (Söllner et al., 2008). The obliquity factor F can be expressed as follows:

$$F = \frac{1}{\cos\theta}, \tag{4.1}$$

where θ is the angle of incidence of each recorded trace. In the time-space (t-x) domain, the equations to reconstruct the up- and down-going parts of the pressure wavefield, denoted P^{up} and P^{down} respectively, can be written as:

$$P^{up} = \frac{1}{2}(P - \rho v F V_z), \tag{4.2}$$

$$P^{down} = \frac{1}{2}(P + \rho v F V_z), \tag{4.3}$$

where P is the total pressure, V_z the total vertical component of particle velocity, ρ the water density, and v the propagation velocity in water. All the variables in equations (4.2) and (4.3) are known apart from F. To determine θ for each single event, we generally apply plane-wave decomposition. Indeed, according to the paraxial ray theory (Ursin, 1982; Hubral, 1983; Bortfeld, 1989) θ can be related to the slowness s of a plane wave as follows:

$$s = \frac{\sin\theta}{v}. \tag{4.4}$$

By rearranging equation (4.4), we can also write:

$$\cos\theta = \sqrt{1-(sv)^2}. \tag{4.5}$$

The frequency-wavenumber *(f-k)* domain is a common representation used in seismic data processing. In this domain, each plane wave at a given frequency maps to a particular wavenumber. Hence, the slowness of a plane wave s can be related to the horizontal angular wavenumber k_x in a straightforward manner:

$$s = \frac{k_x}{\omega}, \tag{4.6}$$

where ω is the angular frequency. From (4.5) and (4.6) we can write:

$$\cos\theta = \sqrt{1-\left(\frac{v}{\omega}k_x\right)^2}, \tag{4.7}$$

with k_z being the vertical angular wavenumber. From equation (4.7) we can also write:

$$\cos\theta = \frac{v}{\omega}k_z, \qquad \text{where:} \qquad k_z = \sqrt{\left(\frac{\omega}{v}\right)^2 - {k_x}^2} \tag{4.8}$$

Consequently, equations (4.2) and (4.3) can be rewritten in the *f-k* domain as follows: (2D case)

$$P^{up}(\omega,k_x) = \frac{1}{2}\left(P(\omega,k_x) - \frac{\rho\omega}{k_z}V_z(\omega,k_x)\right), \tag{4.9}$$

$$P^{down}(\omega,k_x) = \frac{1}{2}\left(P(\omega,k_x) + \frac{\rho\omega}{k_z}V_z(\omega,k_x)\right). \tag{4.10}$$

Equations (4.9) and (4.10) are used to reconstruct the high-frequency component of the up- and down-going pressure wavefield in a dual-sensor acquisition configuration. For the low-frequency component, the particle velocity data can be predicted from the relatively clean pressure data using the following equation (Amundsen et al., 1995):

$$V_z(\omega,k_x) = -\frac{k_z}{\omega\rho}\left(\frac{1+e^{-2jk_zz}}{1-e^{-2jk_zz}}\right)P(\omega,k_x), \tag{4.11}$$

where z is the streamer depth, and j is the imaginary unit such that $j^2 = -1$. This procedure assumes a flat streamer and the sea surface to behave as a free surface. After emulating the low-frequency component of the particle velocity data, the full frequency bandwidth of both pressure and particle velocity data can be used with equations (4.9) and (4.10). However, it is also possible to use processing-based methods to handle the low-frequency component. Day et al. (2013) have demonstrated that emulating the low-frequency particle velocity and then performing wavefield separation is equivalent to deghosting the pressure data using processing-based methods. In the next section, we will describe these methods in more detail.

4.1.2. Processing-based methods for receiver deghosting

Processing-based methods for receiver-side deghosting were originally designed for data recorded with single-sensor streamers, which means that such methods are applied on pressure-only measurements containing both desired primaries and unwanted ghosts. These methods seek to find the time delay τ between the primary and the ghost for each single trace. Let us consider a pressure signal in time $p(t)$, containing only the primary and its corresponding receiver ghost. Such signal can be mathematically expressed as follows:

$$p(t) = r(t) + r_{ss}\ r(t-\tau), \tag{4.11}$$

where $r(t)$ denotes the recorded primary event in time at a given receiver location, and $r(t-\tau)$ the recorded ghost at the same receiver with similar amplitudes but delayed with τ. The reflection coefficient at the sea surface is denoted r_{ss} and is generally approximated as -1. Hence, in the frequency-space *(f-x)* domain, equation (4.11) can be written as:

$$P(\omega) = R(\omega) - R(\omega)\ e^{-j\omega\tau}\ , \tag{4.12}$$

Therefore, we can write:

$$R(\omega) = P(\omega)\ .U(\omega)\ , \qquad \text{where:} \qquad U(\omega) = \frac{1}{1-e^{-j\omega\tau}}. \tag{4.13}$$

where $U(\omega)$ denotes the 2D deghosting operator in the *f-x* domain. The time delay τ is a function of the propagation velocity in water v, the receiver depth z, and the angle of incidence θ (Zhang et al., 2018). This time delay can be geometrically calculated as shown in Figure (4.2), and mathematically expressed as:

$$\tau = \frac{2z}{v}\cos(\theta). \tag{4.14}$$

As in the 4.1.1 subsection, the problem is to find the angle of incidence θ, which is a kinematic attribute of the wavefield related to the slowness of a plane wave (4.4). Hence, plane wave decomposition can again be used to find the correct deghosting operator for each single event. This is generally achieved by applying the deghosting operator U in the frequency-slowness *(f-s)* domain (Masoomzadeh and Woodburn, 2013; Robertsson and Amundsen, 2014; Zhang et al., 2015; Zhang et al., 2018), or in the *f-k* domain (Amundsen, 1993; Robinson and Treitel, 2008; Wang et al., 2017). In these domains τ can be written as (from equations (4.5) and (4.14)):

$$\tau(s) = \frac{2z}{v}\sqrt{1-(sv)^2}, \tag{4.15}$$

or as (from equations (4.8) and (4.14)):

$$\tau(k_z) = \frac{2z}{\omega}\, k_z. \tag{4.16}$$

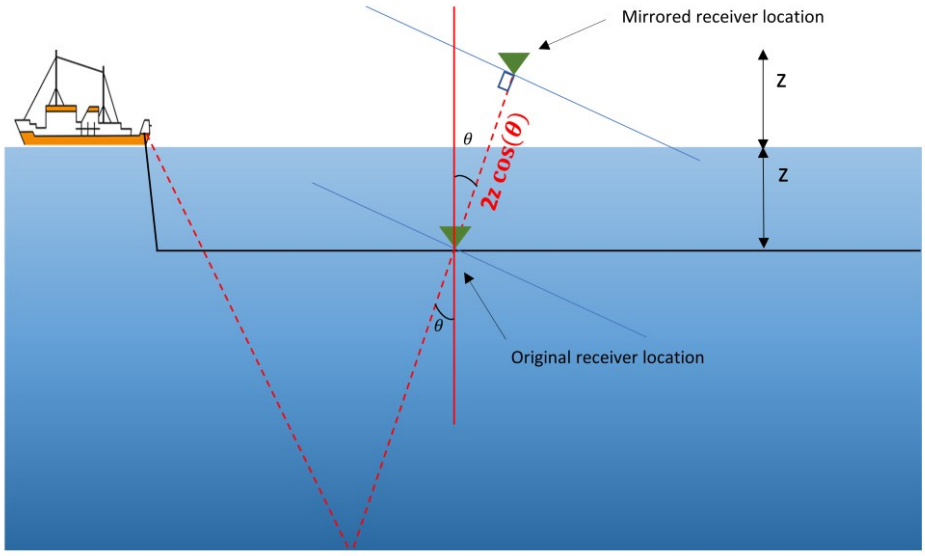

Figure 4.2: *Geometric illustration for calculation of the ghost time delay* τ.

4.2. Operators in a sparse domain characterized by wavefield kinematic parameters

In this thesis our aim is to apply wavefield separation and receiver deghosting processing steps directly in a compressed domain. As discussed in the previous section 4.1, kinematic parameters such as the incidence angle or the slowness of plane wave events are needed when applying these processing steps. Therefore, in this section, we identify simple operators based on these kinematic parameters that can be applied in a sparse domain suitable for compression.

Parabolic Dictionary Learning (PDL), which was introduced in Chapter 3, transforms data to the dictionary learning (DL) sparse domain (Chapter 1) where atoms are constrained to have a parabolic shape. Hence, it can describe seismic events as a linear combination of parabolic elementary waveforms characterized by kinematic parameters of the wavefield such as the slope (or slowness) s and the curvature c. Thus, in the following we investigate how the kinematic parameters can be used in the PDL sparse domain to allow wavefield separation and deghosting processing steps.

4.2.1. Scaling operator in the compressed domain

In the previous sections we have described the wavefield separation process, where particle motion data need to be scaled by an obliquity factor. To do so, plane wave decomposition is commonly used to compute kinematic attributes of the wavefield such as the slowness. Here, we want to investigate the possibility of applying a scaling operator to each trace of each elementary event using kinematic parameters of the PDL sparse domain.

We consider a synthetic example simulating a shot gather containing a family of events with different dips as shown in Figure 4.3-a. The data consists of 70 traces sampled at 10 m and 256 time samples sampled at 4 ms. In this figure, the blue frame depicts a first patch of 64 samples and 70 traces containing a single event, and the red frame depicts a second patch of the same size containing 4 events with conflicting dips. Let us assume that our objective in this experiment is to correct for the obliquity scaling problem in these patches. First, we apply PDL to the shot gather. The number of learned atoms was set to 2000, where each atom is of size 64 by 70. Each patch was constrained to be represented with 4 atoms. Figure 4.3-b illustrates the decomposition of the patch depicted with blue frame using PDL. From this decomposition we can see that the original patch denoted $\mathbf{y_1}$ has the same shape as the atoms $\{\mathbf{d}_k\}_{k=1}^{k=4}$, and hence the same slope and curvature. Additionally, we know that each $\mathbf{d}_k$ is characterized by a slope s_k (which is the same as the slowness), a curvature c_k,

and a reference receiver location corresponding to a reference trace x_r^{ref}. Hence, by taking the derivative of equation (3.3), we can find the slope s_k^i of each trace i in an atom k, and thus we can write:

$$s_k^i = 2c_k \left(x_r^i - x_r^{ref}\right) + s_k \tag{4.17}$$

Now that we have access to the slope of each single trace in an atom $\mathbf{d}_k$, we can easily find the angle of incidence θ_k^i of each single trace i of that atom via (4.5) and thus the obliquity factor F_k^i characterizing each trace of the atom via (4.1).

In the example illustrated in Figure 4.3-b, where $\mathbf{y_1}$ is expressed as a linear combination of the atoms $\{\mathbf{d}_k\}_{k=1}^{k=4}$ such that $\mathbf{y_1} = \sum_{k=1}^{k=4} \alpha_k \mathbf{d}_k$, a given trace i located at a specific location x_r^i has the same scaling factor F^i for all atoms because they all have the same slope and curvature. Hence, multiplying all the traces i in the different atoms by F^i is the same as scaling the trace i of the patch $\mathbf{y_1}$ by F^i. This should lead to the reconstruction of the scaled version of the patch $\mathbf{y_1}$ after obliquity correction, given that each trace of the patch $\mathbf{y_1}$ has the same slowness as its corresponding trace in the different atoms.

In the example depicted in Figure 4.3-c, $\mathbf{y_2}$ is also expressed as a linear combination of the atoms $\{\mathbf{d}_k^{'}\}_{k=1}^{k=4}$ such that $\mathbf{y_2} = \sum_{k=1}^{k=4} \alpha_k^{'} \mathbf{d}_k^{'}$. However, in this example the patch $\mathbf{y_2}$ contains events with conflicting dips. Hence, we need to have access to the kinematic parameters of each elementary event in order to find the scaled version of this patch. Indeed, from this figure PDL succeeds in decomposing the patch into the different elementary waveforms from which it is composed. Therefore, scaling the same way as in the example depicted in Figure 4.3-b, will reconstruct the desired scaled patch. These examples show that PDL permits the application of scaling operators to elementary events based on their kinematic attributes. Hence, PDL can be used on a larger scale with vertical particle motion data to correct for the obliquity scaling problem.

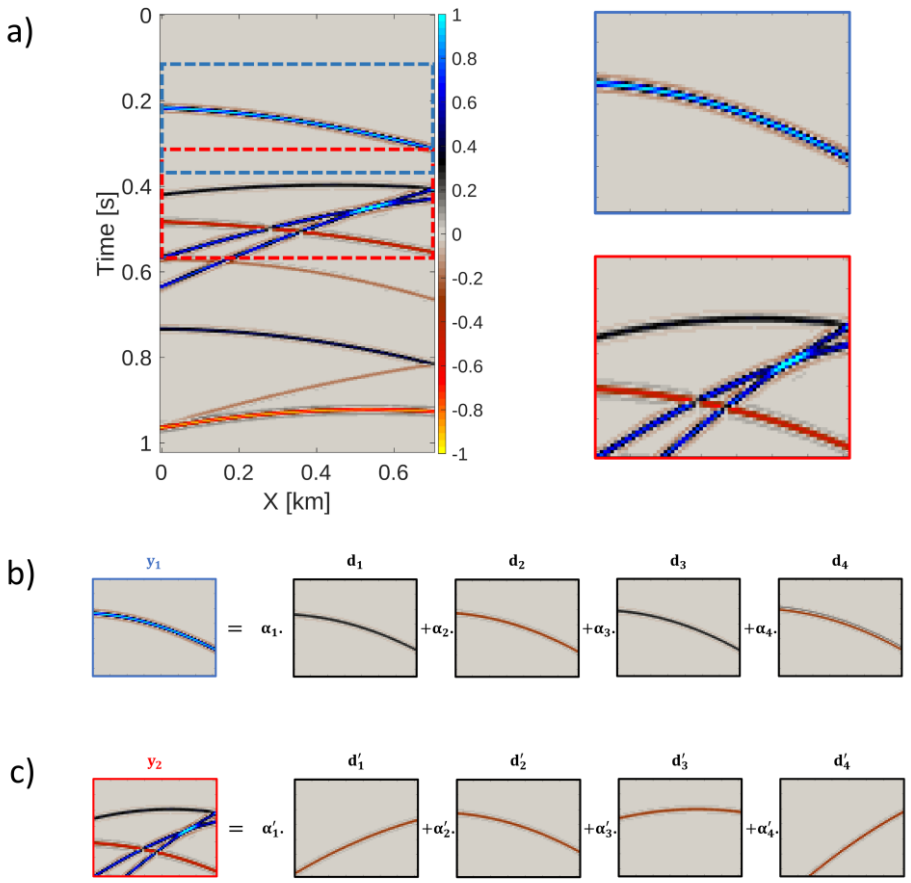

Figure 4.3: *Synthetic example with many dipping events. a) shot gather depicting two patches with blue and red frames. b) Decomposition of a single event patch to elementary parabolic waveforms using PDL. d) Decomposition of a multiple-events patch to elementary parabolic waveforms using PDL.*

4.2.2. Frequency-space operator in the compressed domain

In the case of processing-based methods for receiver ghost removal, and as discussed in the previous section, we need to multiply pressure data with a deghosting operator U defined in the *f-x* domain (4.13). This operator contains an unknown variable τ. Thus, plane-wave decomposition was needed to determine τ for each single event. Here, we would like to investigate the feasibility of applying such an operator using PDL. We know from the previous subsection that PDL gives access to the slowness of each trace in the parabolic atoms using (4.17). Hence, from (4.5) and (4.14) PDL is able to find the time delay τ of the ghost for each single trace in the parabolic atoms. Yet, applying the operator U as in equation (4.13) implies transforming the atoms from the *t-x* domain to the *f-x* domain. Indeed, if we consider a patch denoted $\mathbf{y}$, in the PDL sparse domain this patch can be mathematically expressed as $\mathbf{y} = \sum_{k=1}^{k=\mathrm{K}} \alpha_k \, \mathbf{d}_k$, where $\{\mathbf{d}_k\}_{k=1}^{k=\mathrm{K}}$ denotes the K parabolic atoms and α_k the non-zero coefficients in a sparse vector. Hence, given that the FFT (Fast Fourier transform) is a linear operator, we are able to write the equation in the *f-x* domain as:

$$\mathbf{y}(\omega) = \alpha_1 . \mathbf{d}_1(\omega) + \alpha_2 . \mathbf{d}_2(\omega) + \cdots + \alpha_k . \mathbf{d}_{\mathrm{k}}(\omega) \ . \tag{4.18}$$

Multiplying both sides with the operator U gives:

$$U(\omega) . \mathbf{y}(\omega) = \alpha_1 . U(\omega) . \mathbf{d}_1(\omega) + \alpha_2 . U(\omega) . \mathbf{d}_2(\omega) + \cdots + \alpha_k . U(\omega) . \mathbf{d}_k(\omega) \tag{4.19}$$

This means that multiplying the atoms by a given operator in the *f-x* domain is supposed to give the same result as multiplying the patch by that operator. Of course, in the case of multiple conflicting dips in a patch as in Figure 4.3-c, different operators will be applied to the different elementary events similarly to the experience we have previously conducted with the scaling operator.

Even though deghosting the different atoms leads to a similar result as deghosting the different elementary events of a patch, removing the ghost from the whole recorded signal is not guaranteed. Indeed, some patches may not contain the primary and its ghost, particularly if an event is captured at the edge of the patch, resulting in an inaccurate deghosting operation at these edges. To better understand this problem, let us consider a simple experiment using a 1D trace. Figure 4.4 displays a 1D trace containing both the primary and its ghost (in blue). The trace is a synthetic trace of 512 samples with a time sampling of 2 ms, where τ is known. Hence, this trace can be easily deghosted (in red) using equation (4.13) to reconstruct the primary event (in black). However, if we decompose this signal into non-overlapping or overlapping

patches, and then apply the operator on each individual patch we will not end up with a similar result. Figure 4.5 illustrates the non-overlapping case, where we can see that if a patch contains only parts of the primary or parts of the ghost, the deghosting operator creates artefacts and hence, when the patches are concatenated, the result does not correspond to that in Figure 4.4. The conventionally used work-around to this problem is to extract overlapping patches and apply a taper, such as the Hamming function, to the edges of the patches which will avoid deghosting edges of the patches containing only parts of the primary or the ghost. Additionally, an overlap of at least 75% is recommended when the FFT is applied on Hamming tapered patches (Zhivomirov, 2019). Figure 4.6 illustrates the overlapping and tapered case. In fact, applying an overlap of 75% results in the extraction of 13 overlapping patches. Nevertheless, in the figure we show only nine patches since no information is contained in the two first and the two last patches. We can now see that, when applying the deghosting operator on each patch, the reconstructed signal is much smoother. After summing and averaging we obtain the deghosted signal. Consequently, PDL can be used on a larger scale to deghost pressure data.

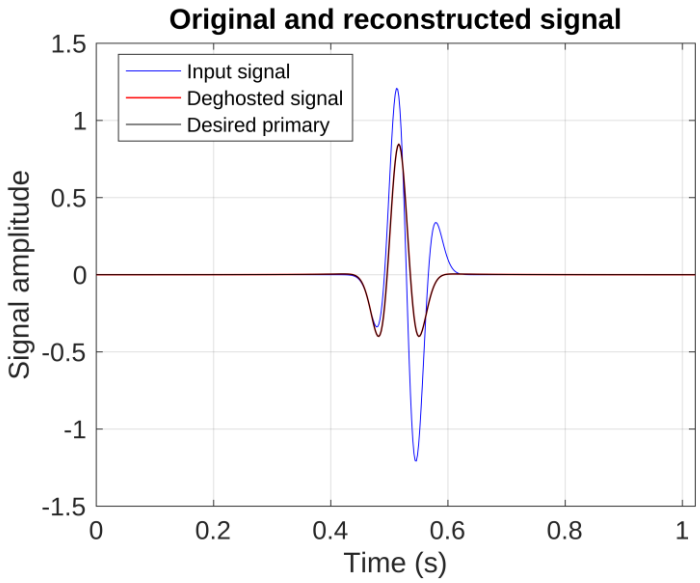

Figure 4.4: *Single trace containing both the primary and the ghost depicted in blue. The primary alone is depicted in black. Deghosting the input trace in blue using equation (4.13) results in the deghosted signal in red.*

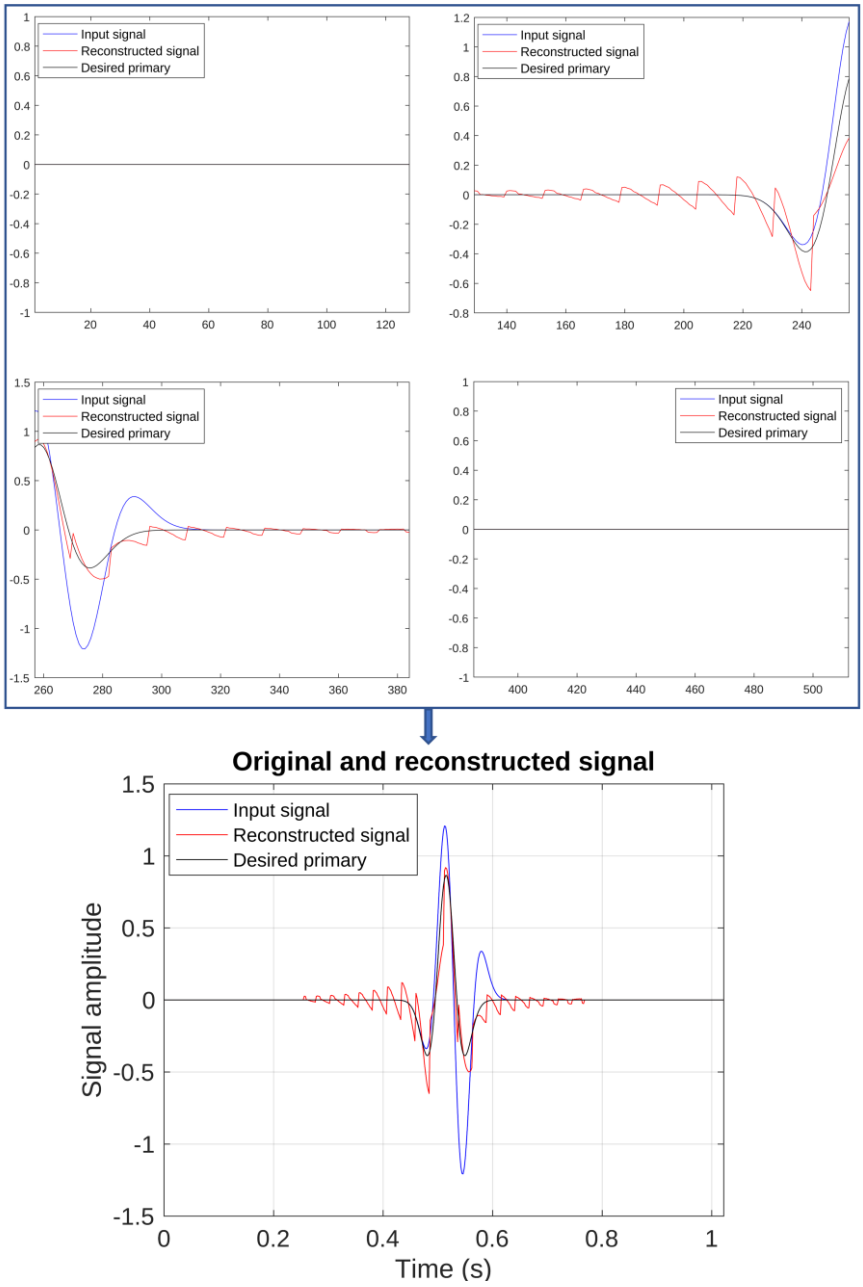

Figure 4.5: *Experiment where the original signal displayed in Figure 4.4 is split into four non-overlapping patches, before applying the deghosting operator on each patch separately.*

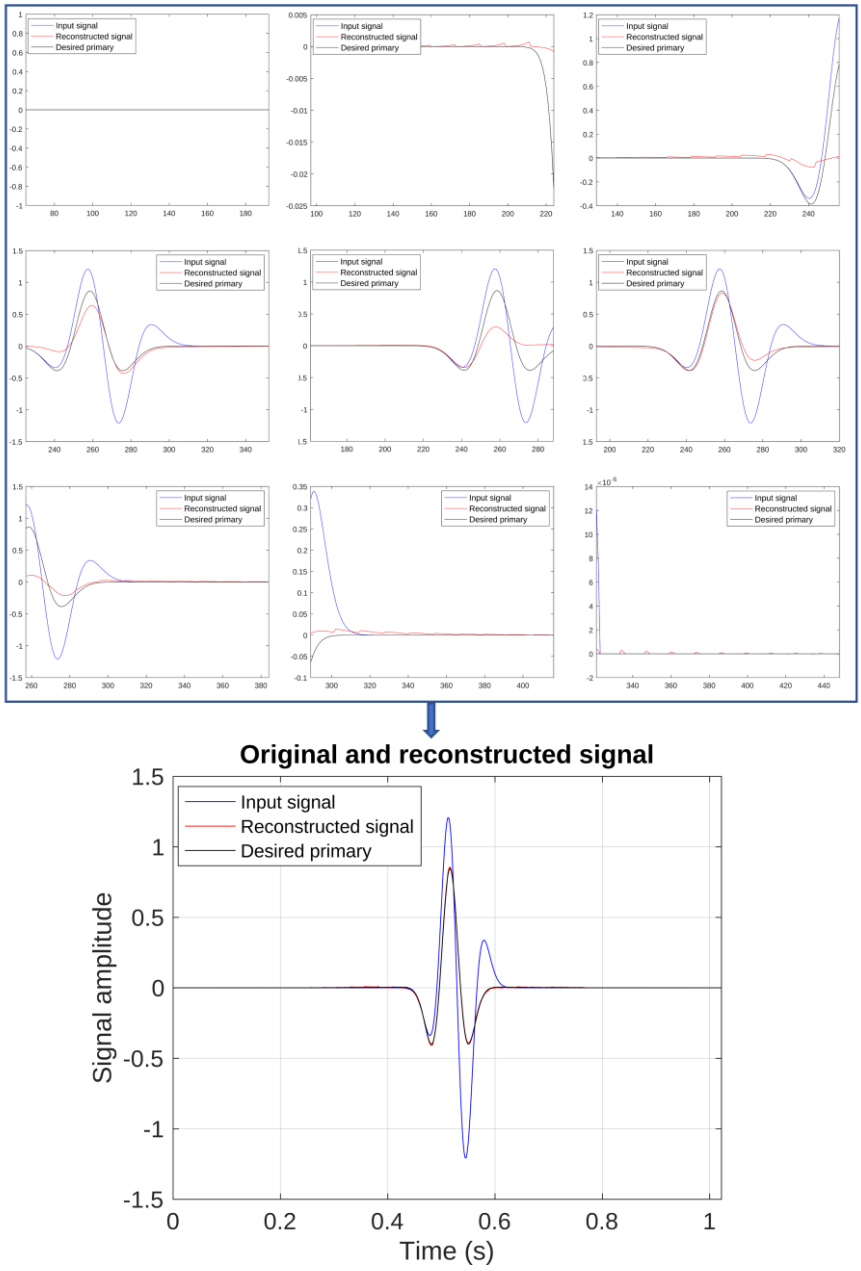

Figure 4.6: *Experiment where the original signal displayed in Figure 4.4 is split into thirteen overlapping patches. Each individual patch is tapered before applying the deghosting operator. Finally, the overlapping patches are added and averaged.*

Chapter 5

Main Scientific Contribution

5.1. Paper I

A dictionary learning method for seismic data compression

Mohammed Outhmane Faouzi Zizi and Pierre Turquais.

Geophysics, Volume 87, No. 2 (2022); P. V101-V116.
DOI: 10.1190/GEO2020-0948.1

Motivation

In marine seismic exploration, surveys can generate vast amounts of data, often several terabytes, making the storage and transfer of these data sets costly and challenging. Hence, developing a compression method that could reduce the data size while reaching a high-quality reconstruction would be highly beneficial for the industry. Seismic data sets are highly redundant, and traditional compression methods based on predefined transforms do not fully exploit this redundancy. As a result, there has been growing interest in applying data-driven methods such as dictionary learning (DL) to compress seismic data. This motivated us to develop an efficient DL-based compression algorithm specifically designed for seismic data. The algorithm can fully exploit the redundancy of information in seismic data and achieve high compression ratios (CRs) while preserving the quality of the seismic signals at the desired level of signal-to-residual error ratio (SRR).

Key contributions and findings

In this paper, we developed a DL-based compression workflow which learns small-sized sub-dictionaries from local windows of multiple seismic shot gathers. This process captures the local similarities between the seismic shot gathers and enhances sparsity. The proposed method achieves superior performance in terms of CR and SRR when compared to conventional compression methods such as the zfp software or algorithms from the Seismic Unix package. The capability of the workflow to preserve the seismic signal was evaluated for different applications. For instance, when the workflow is used for near-real-time data

transmission or long-term data storage, we observed insignificant signal leakage on a 2D line stack while the DL method reached a CR higher than 20. For other applications such as visual quality control (QC) of shot gathers, the method preserved the visual aspect of the data even when a CR of 95 is reached. These findings suggest that the proposed DL-based compression workflow can significantly reduce the cost associated with storing and transferring large seismic data sets while preserving the seismic signal at the desired level of SRR.

5.2. Paper II

Dual-sensor wavefield separation in a compressed domain using parabolic dictionary learning

Mohammed Outhmane Faouzi Zizi, Pierre Turquais, Anthony Day, Morten W. Pedersen, Leiv J. Gelius.

Geophysical Prospecting, (2023); P. 1-19. DOI: 10.1111/1365-2478.13348

Motivation

The marine seismic industry faces numerous challenges, from data acquisition to imaging, due to the massive amounts of seismic data obtained from marine surveys. One effective way to reduce storage and data transfer costs is to use data compression algorithms at the early stages of seismic processing. Faouzi Zizi and Turquais (2022) have recently developed a dictionary learning (DL) method, which has been shown to provide state-of-the-art results for seismic data compression. However, compressing seismic data with this method, or other conventional methods, does not offer benefits throughout the entire processing time as seismic data must be decompressed before processing or imaging. This motivates the need to enable a seismic processing step directly in the compressed domain, specifically the dual-sensor wavefield separation processing step, which is typically applied early in the seismic processing sequence.

Key contributions and findings

In this study, we successfully implemented a novel method for dual-sensor wavefield separation in a compressed domain using a parabolic dictionary learning algorithm (Turquais et al., 2018). This method, referred to as WSPDL, transforms the input data sets, i.e., pressure and vertical particle velocity measurements, to a compressed domain with two main components: a dictionary of parabolic atoms and a set of sparse coefficients. The atoms in the dictionary

are characterized by kinematic parameters such as slope and curvature, which enable proper correction of obliquity scaling directly in the compressed domain. This correction allows the up- and down-going parts of the pressure wavefield to be separated successfully. The performance of the WSPDL method was tested and validated by quantifying the accuracy of the up-going pressure field reconstruction using the signal-to-residual ratio (SRR) metric, based on a 100-shot synthetic data example, where the true result was known. The study also demonstrated similar wavefield separation performance compared to an optimized industry-standard FK based method on both synthetic and field data examples. The WSPDL method was also shown to be robust with respect to spatial aliasing while reaching a high data compression rate of more than 15, which enables data transfer from vessels to onshore processing centers. The transmitted compressed data can be used to reconstruct, with a simple matrix multiplication, not only the input data sets recorded by hydrophones and geophones, but also the up- and down-going parts of the wavefield.

5.3. Paper III

Low frequency seismic deghosting in a compressed domain using parabolic dictionary learning

Mohammed Outhmane Faouzi Zizi, Pierre Turquais, Anthony Day, Morten W. Pedersen, Leiv J. Gelius.

Submitted to *Geophysical Prospecting*.

Motivation

Acquisition-based methods such as dual-sensor or multicomponent towed streamers, have shown to provide state-of-the-art results for removing receiver-ghosts from seismic data. However, low frequencies of the recorded vertical particle motion data are generally heavily contaminated with noise. Hence, processing-based methods are used to deghost the low frequency component of the recorded pressure data. Moreover, legacy data are recorded using conventional single-sensor streamer acquisition and also rely on processing-based methods to remove the receiver-side ghost. Faouzi Zizi et al. (2023) have introduced a method to enable dual-sensor wavefield separation directly in a compressed domain using parabolic dictionary learning (WSPDL). This motivated us to develop a novel method, referred to as LFD-PDL, to deghost receiver-side low frequency pressure data directly in the DL compressed domain.

Key contributions and findings

The LFD-PDL method first decomposes the low-frequency pressure data into a set of sparse representations and a dictionary of parabolic atoms characterized by local kinematic parameters. The local kinematic parameters are used to calculate the ghost time delay associated with each trace of the parabolic atoms, and the deghosting operator is applied to each trace independently. The reconstructed ghost-free pressure data is obtained using the deghosted atoms. The proposed method was evaluated using a synthetic data set where the true ghost-free data was known. The accuracy of the method was assessed using visual inspection and the signal-to-residual ratio (SRR) metric. Comparisons with an optimized industry-standard FK method were carried out using both synthetic and field data sets. The findings suggested that the LFD-PDL method can achieve deghosting performance comparable to that of the FK method, while also providing the benefit of being applicable directly in the DL compressed domain with a data compression rate (CR) greater than 5. Additionally, the LFD-PDL method can be easily generalized to high-frequency pressure data, and thus can be used to deghost the full frequency bandwidth of data recorded using conventional single-sensor streamer acquisition.

Chapter 6

Discussions and outlook

6.1. Discussions and conclusions

6.1.1 Amplitude balance

In this subsection we discuss the impact of balancing amplitudes between shallow and deep parts on compression results. In Papers I, II, and III, DL (Dictionary Learning) and PDL (Parabolic Dictionary Learning) are applied with the "*error constraint mode*". This mode is designed to better reconstruct data with higher amplitudes. Therefore, it relies on an appropriate gain applied to the data prior to compression. For example, if no amplitude gain is applied before applying DL or PDL, events in the deep part of the model will not be reconstructed with the same accuracy as the shallow part. That is because seismic events in the deeper part tend to be more attenuated than those in the shallower part. Hence, balancing amplitudes is a crucial step before applying seismic data compression. To balance seismic amplitudes, it is common to apply a time-based gain function. For instance, in Paper I a T-square spherical divergence correction was applied to the seismic shot gathers before compression/decompression. It consists of multiplying the amplitude of each sample by the square of its corresponding recording time. Later, this amplitude gain was removed before processing the data and obtaining a 2D stack section. The purpose of the applied gain was to preserve low-amplitude signals in the deep part of the seismic data set.

Consider again the same data set used in the "Transmission and long-term storage" section of Paper I. There, the data set was used to compare "harsh", "medium" and "soft" compression levels. In each of the three cases the T-square spherical divergence was applied. Now, we use the same data set to compare different time gains, namely: T-square spherical divergence, T spherical divergence, and no spherical divergence, denoted T2, T1, and T0, respectively. Now, in this experiment we use the so called "harsh" compression, defined in Paper I. This compression level was chosen to better assess the visual differences between the different time gains. Figure 6.1-a displays the original 2D stack section when no compression is applied, then Figures 6.1-b-d display the corresponding residuals after applying T0, T1 and T2, respectively, to shot gathers prior to "harsh" compression/decompression. From Figure 6.1-b, we

observe that, when no spherical divergence (T0) is applied, some of the high amplitude events characterizing the blended energy in the deep part are removed. When T1 is applied (Figure 6.1-c), the residuals seem to be well balanced. Finally, when T2 is applied (Figure 6.1-d) the deep part is better preserved than the shallow one. Even though T1 seems to provide the best visual results with this data set (Figure 6.1-c), the T2 spherical divergence was still preferred in Paper I because it ensured the reconstruction of low amplitude signals around the high amplitude blended energy. Overall, the choice of time gain should be user defined and one should carefully pick the time gain to apply prior to the DL-based compression depending on the desired output.

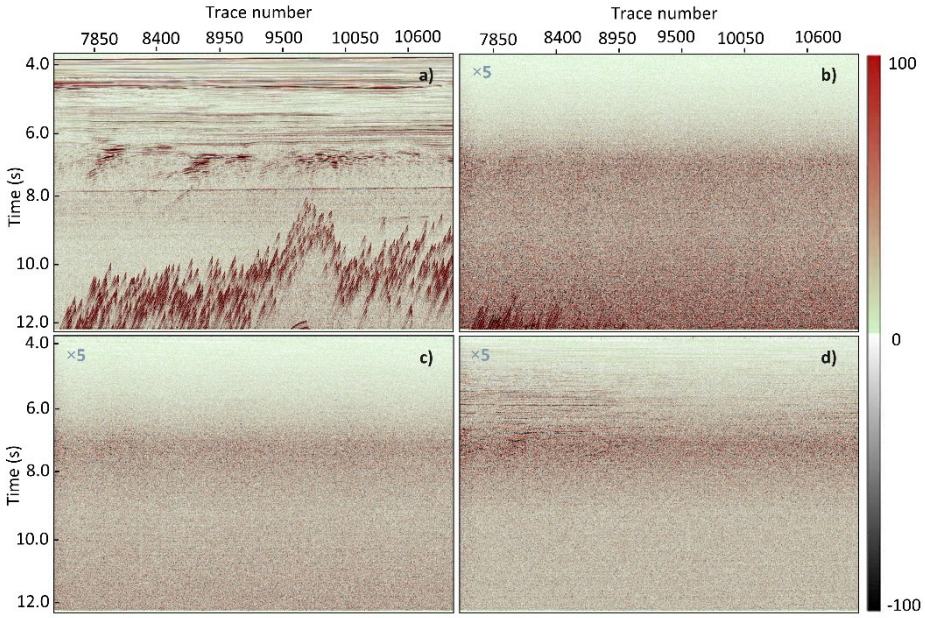

Figure 6.1: *a) 2D stack section similar to the one displayed in Paper I, Figure 11-a. b) c) and d) are corresponding residuals after applying "harsh" compression/decompression to shot gathers which amplitudes are gained with T0, T1, and T2, respectively.*

A similar time gain can also be applied prior to the WSPDL (Wavefield Separation using Parabolic Dictionary Learning) method developed in Paper II. There, the same time gain should be applied to both pressure and velocity data. However, in Paper III we apply the deghosting operator on pressure data only. Considering a single trace containing both the primary and the ghost, the two events are not localized at the same time positions. Hence, applying a time gain

such as T1 or T2 prior to LFD-PDL (Low Frequency Deghosting using Parabolic Dictionary Learning) will result in applying different gains to the primary and the ghost. This would lead to the wrong deghosting operator being applied. To balance amplitudes when LFD-PDL is used, we have implemented a fixed time gain in the method presented in Paper III. The process involves a data set consisting of L patches represented by $\{\mathbf{y_i}\}_{i=1}^{L}$ where each patch $\mathbf{y_i}$ is multiplied by a fixed scalar, either t_j or t_j^2 (depending on the user's choice), where j denotes the number of patches extracted in the time dimension. These scalars are determined by the time position from where $\mathbf{y_i}$ was extracted. This multiplication occurs prior to applying OMP to identify the sparse representations $\{\mathbf{x_i}\}_{i=1}^{L}$. The value of t_j depends on the time position. Hence, OMP is solved for each $t_j\mathbf{y_i}$. Later, the deghosting operator is applied to the parabolic atoms and the sparse representations $\mathbf{x_i}$ are quantized and coded. After decoding and before reconstruction, each $\mathbf{x_i}$ can be divided by the corresponding t_j, since each $\mathbf{x_i}$ corresponds to a given $\mathbf{y_i}$. This allows for amplitude balancing before transitioning to the DL domain and at the same time ensure that the deghosting operator is applied on the original amplitudes.

6.1.2 Dictionary learning parametrizations

In Paper I we have developed a DL-based algorithm specifically designed to compress seismic data. One of the main characteristics of that algorithm was its capacity to fully exploit similarities between shot gathers. This algorithm was based on learning localized sub-dictionaries for different time and offset positions. Yet, each of these sub-dictionaries was learned from the many similar shot gathers in the data set. Once a dictionary was learned from a local window, it was used to only reconstruct events localized in that specific window for all shot gathers. Hence, such an extraction strategy led to sparser representations given that similar waveform patterns are locally repeated many times over the different shot gathers around the same t-x locations. Figures I-1 and I-2, which refer to Figures 1 and 2 in Paper I respectively, summarize the different steps of the DL-based method parametrization and extraction strategy. However, even though this strategy was promoted in Paper I, it was not used with PDL in Papers II and III, where a single dictionary was learned for a set of gathers. In this subsection, we discuss the motivation behind this change of strategy and parametrization.

Let us consider the simple field data example of 100 shot gathers introduced in the methodology section of Paper I. Shot gathers of this data set are displayed in Figure I-1. Let us recall the characteristics of this data set and the parametrization of the DL-based algorithm used there. The data are sampled at 4 ms in time and 12.5 m in space. The data set comprises $Nt = 1251$ time

samples, and $Nx = 200$ traces. Using the DL-based method developed in Paper I, 25 dictionaries are learned for the whole data set: one dictionary for each window of size one second by 500 meters. Each of these windows contain $M = 15500$ 2D patches of size eight time samples by eight space samples. Hence, in each local window all the extracted M patches are used as a training set to learn a sub-dictionary representing 256 atoms also of size eight by eight. The 25 learned sub-dictionaries are later used to reconstruct the M patches for each window. In that experiment we were targeting a signal-to-residual ratio (SRR) of 20 dB (equation I-3, which refers to equation 3 in Paper I). Figure 6.2-a displays one shot gather from this seismic data set where no compression has been performed. Figure 6.2-b displays the residuals after reconstructing this shot gather with the DL-based method, and subtracting it from Figure 6.2-a. The residuals appear as random noise uniformly distributed across the shot gather. Now, we run a second experiment where we apply the same parametrization and extraction strategy, but with PDL instead of DL. Figure 6.2-c depicts the corresponding residuals. From this figure, we observe that the Orthogonal Matching Pursuit (OMP) algorithm (Chapter 2) did not succeed in reaching the desired SRR of 20 dB while reconstructing windows with dictionaries learned via PDL. For instance, the window located at 2 s – 3 s and 2 km – 2.5 km was reconstructed with SRR of 7 dB instead of 20 dB. When applied on local regions where nearly all waveforms have the same kinematic attributes PDL seems to be less stable than DL. For example, in the case of the specific window depicted above, applying PDL has led to learning parabolic atoms having the same slopes and curvatures. Consequently, as soon as a patch was deviating slightly from the learned kinematic characteristics, OMP was not able to reconstruct it using atoms that were all constrained to have a different slope and curvature. One way to stabilize PDL was to provide it with training patches from the entire gathers, and thus force it to learn atoms with different kinematic parameters. Finally, we run another experiment where only one parabolic dictionary of 6000 atoms is learned (as in Papers II and III) for the entire shot gathers. Here, the number of dictionary atoms was fixed to 6000 because we used to learn 25 dictionaries of 256 atoms which is equivalent to 6400 atoms learned from the whole data set. Figure 6.2-d depicts the corresponding residuals. The desired SRR level is now reached. However, we can still notice from this figure small discontinuities between the reconstructed patches when PDL is used. That is the reason behind considering overlapping patches in Papers II and III.

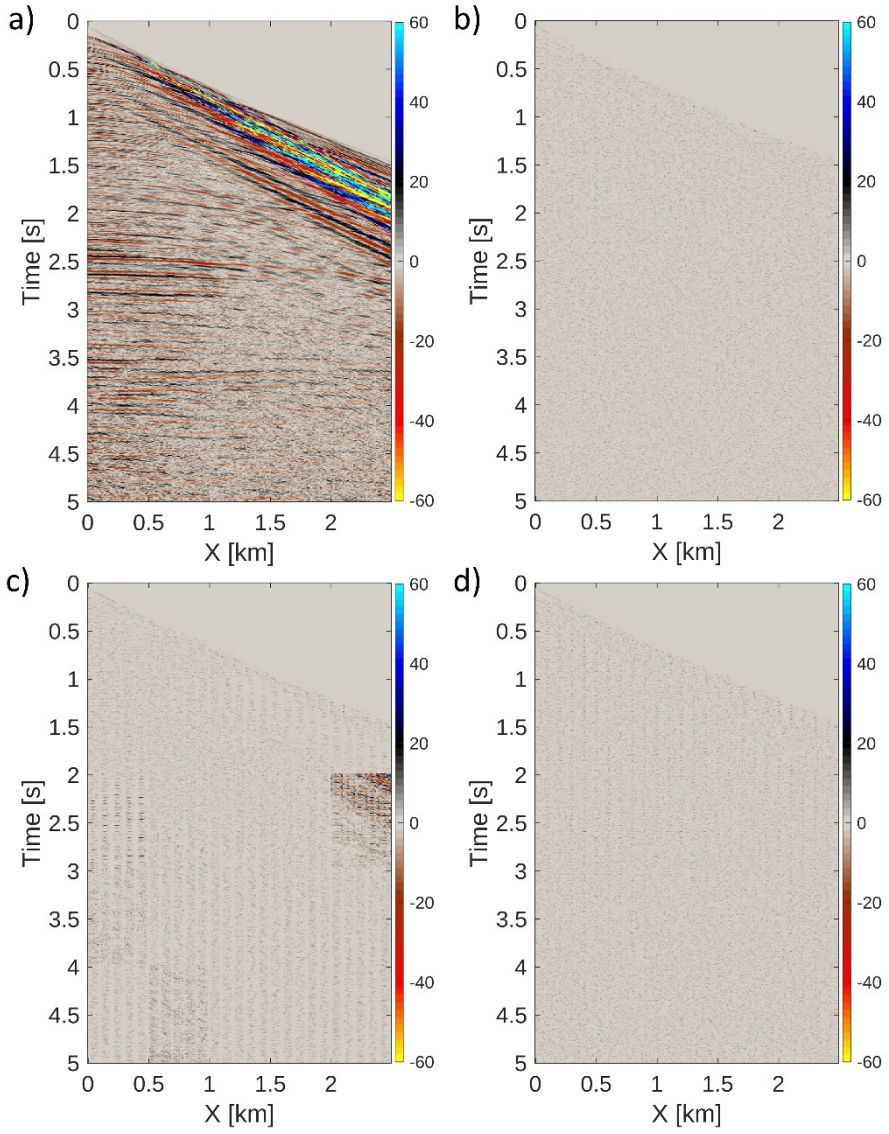

Figure 6.2: *a) Arbitrary shot gather chosen from the 100 shots seismic data set presented in the methodology section of Paper I. b) Residuals after subtracting the reconstructed gather using the DL based method described in Paper I. c) Residuals after subtracting the reconstructed gather using PDL with the parametrization described in Paper I. d) Residuals after subtracting the reconstructed gather using PDL with one dictionary learned from non-localized patterns of all shot gathers.*

6.1.3 Workflow combining wavefield separation and receiver deghosting in the DL compressed domain

In this section, our aim is to integrate the methods developed in Papers I, II, and III into a single workflow and evaluate its advantages over the conventional approach of compression/decompression before processing. We describe the different steps involved in this proposed workflow and discuss its effectiveness in tackling the challenges mentioned in Chapter 1 of this thesis. Our discussion highlights the key findings and outcomes of the research and provides insights into the potential application of this workflow in the marine seismic industry.

In Papers II, and III we have generated a synthetic data set from the same 2D P-wave velocity model (Figures II-5 and III-4). This data set was used in case of both the WSPDL and LFD-PDL methods. We consider the same model in this subsection, which implies a data set example where both the total vertical component of particle velocity and the total pressure wavefield were recorded by geophones and hydrophones, respectively. The recorded pressure and velocity data sets denoted P and V_Z, comprise $N_s = 100$ shots, $N_t = 1856$ time samples, and $N_x = 480$ traces, and are sampled at 2 ms in time and 12.5 m in space, respectively. The first step in our workflow is to split the P data into high frequency (typically above 20 Hz) and low frequency (typically below 20 Hz) components, denoted P_H and P_L, respectively. Also, we only keep the high frequency component of V_Z data since the low frequencies are considered to be highly contaminated with noise. The P_L data are then resampled from 2 ms to 16 ms. As mentioned in Paper III, all information can be recovered from the data at 16 ms due to the fact that a time interval of 16 ms permits representation of frequencies up to 31.25 Hz, which exceeds the 20 Hz of P_L data. Then, we apply WSPDL to P_H and V_Z, and apply LFD-PDL to P_L, as described in Papers II and III, respectively. Figure 6.3 shows the three data sets in the time domain and in the DL compressed domain. This figure combines Figures II-2 and III-1, where patches with yellow, blue, and red frames in the time domain are represented as a matrix multiplication of sparse representations and parabolic atoms with frames of similar colors. More details regarding the relationship between the time domain and the DL compressed domain can be found in Papers II and III. Similar DL parameters to those described in the papers were used here. The number of extracted training patches for each of the 100 shot gathers was set to $M = 10.000$. The targeted SRR was set to 30 dB. The number of learned atoms for each dictionary is $K = 6000$ atoms. The size of the atoms was set to 64 time samples by 8 traces for WSPDL and 16 by 16 for LFD-PDL. That is because 128 ms by 100 m was enough to capture the parabolic moveout with high frequency data (Paper II). While 256 ms by 200 m was needed for LFD-PDL to achieve good deghosting performance (Paper III).

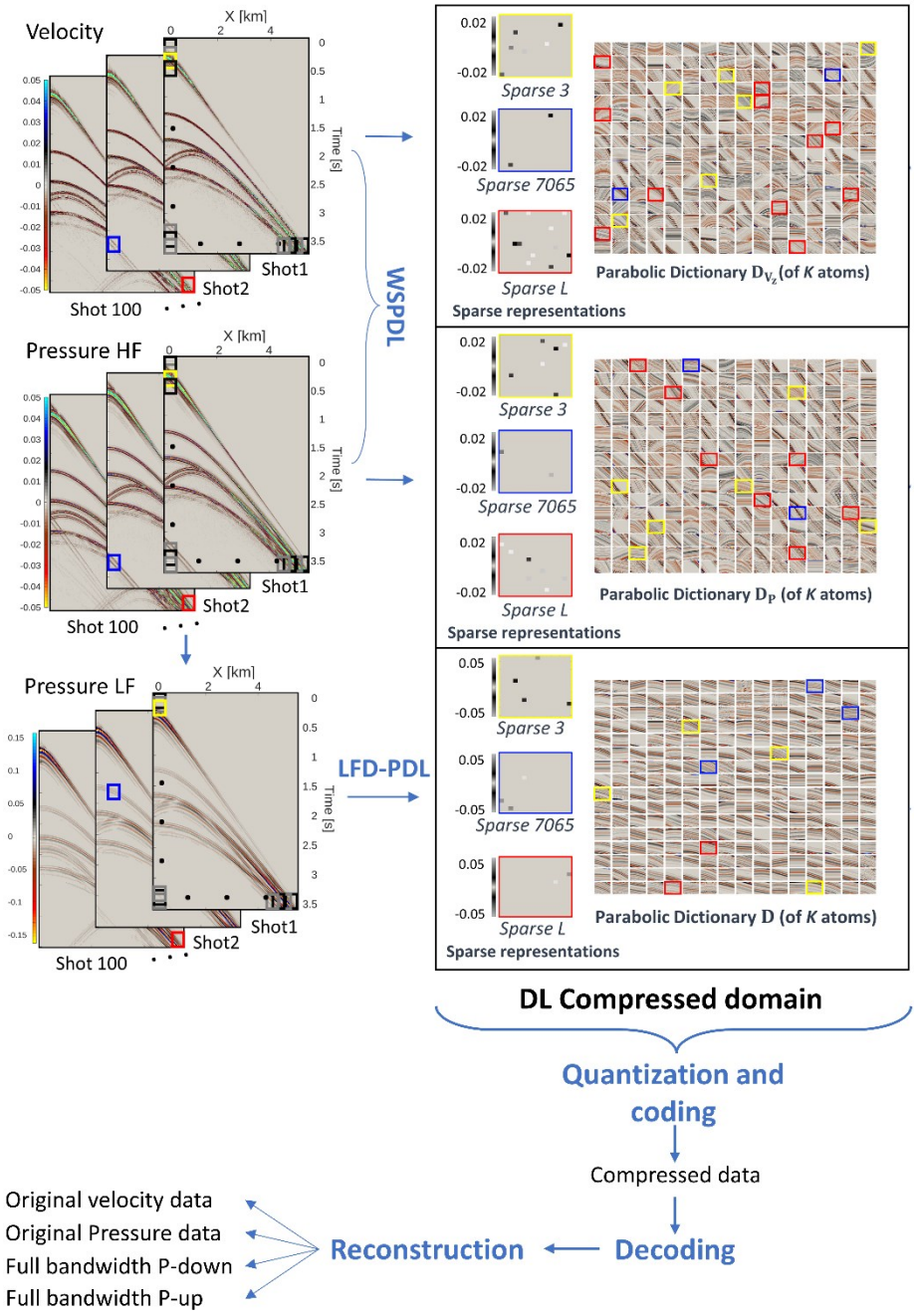

Figure 6.3: *Illustration of the workflow combining dual-sensor wavefield separation and low frequency receiver deghosting in the DL compressed domain.*

Once the data are represented in the DL compressed domain, a scaling operator is applied to each trace of the dictionary atoms learned from V_Z and denoted $\mathbf{D}_{V_Z}$. That is to obtain the dictionary $\mathbf{D}_{CV_Z}$ and correct for the obliquity scaling as described in Paper II. Atoms of the dictionary $\mathbf{D}$ obtained from P_L data are tapered and deghosted to obtain the dictionary $\mathbf{D}_u$ as described in Paper III. Later, $\mathbf{D}_{CV_Z}$, $\mathbf{D}_P$, and $\mathbf{D}_u$ are quantized and coded along their corresponding sparse representations to obtain the compressed data as described in Paper I. These data can then be sent from the vessel to processing centers. Considering each of the recorded P and V_Z data sets to be of size S, the total size of the recorded data is $2S$. Hence, since we apply WSPDL and LFD-PDL on the same synthetic data sets used in Papers II and III, the compression results are similar to the ones presented in the papers. The compressed data after WSPDL are of size $2S/13$ and after LFD-PDL are of size $S/60$. This means that the compressed data are approximately of size $S/5.86$, which is over 11 times smaller than the size of the original recorded data. Note that such compression results are obtained while wavefield separation and deghosting processes reach similar results compared to optimized industry-standard methods as demonstrated in Papers II and III.

Once the data are received by a processing center, the user can choose which data set to recover, the recorded pressure data P, the recorded vertical velocity data V_Z, the full-bandwidth up-going pressure wavefield P^{up}, or the full-bandwidth down-going pressure wavefield P^{down} (Figure 6.3). To do so, the compressed data are first decoded and again expressed in terms of the dictionaries $\mathbf{D}_{CV_Z}$, $\mathbf{D}_P$, $\mathbf{D}_u$, and their corresponding sparse representations. To reconstruct the P^{up} data, the high frequency component P_H^{up} is first reconstructed by simple dictionary combinations (using $\mathbf{D}_{CV_Z}$ and $\mathbf{D}_P$), and matrix multiplication. Paper II describes the details of this process. The low frequency component pressure data P_L^{up} is similarly reconstructed by matrix multiplication of $\mathbf{D}_u$ and its corresponding set of sparse representations. Details of this process can be found in Paper III. Later, the two data sets (P_L^{up} and P_H^{up}) can be simply summed after resampling the resulting P_L^{up} from 16 ms to 2 ms. Note that such resampling can be applied directly in the DL compressed domain by up-sampling each trace of each atom independently. To reconstruct P^{down}, the high frequency and low frequency components P_H^{down}, and P_L^{down} are also reconstructed separately. P_H^{down} is also reconstructed from the combination of $\mathbf{D}_{CV_Z}$ and $\mathbf{D}_P$, and via matrix multiplication with their corresponding sparse representations (Paper II). P_L^{down} reconstruction was not described in Paper III but can easily be obtained by applying the inverse of the deghosting operator to $\mathbf{D}_u$ to reconstruct $\mathbf{D}$. Then, by subtracting $\mathbf{D}_u$ from $\mathbf{D}$ we can obtain $\mathbf{D}_g$, which is the dictionary characterizing the ghost events. Multiplying this dictionary $\mathbf{D}_g$ by the corresponding set of sparse representations gives P_L^{down}. Summing

P_H^{down} and P_L^{down} reconstructs the full bandwidth down-going pressure wavefield data P^{down}. To reconstruct the original pressure data, we reconstruct its high frequency component P_H by multiplying $\mathbf{D}_P$ and the corresponding set of sparse representations (Paper II). Then the low frequency component P_L is obtained by multiplying $\mathbf{D}$ and the corresponding set of sparse representations. Later, they are summed to reconstruct P. Moreover, V_Z can also be reconstructed by reconstructing $\mathbf{D}_{V_Z}$ from $\mathbf{D}_{CV_Z}$ .This implies the application of the inverse scaling operator as described in Paper II. These reconstructions are expected to be computationally cheap since they rely on simple operations such as summations, subtractions, and matrix multiplications. We will later discuss the computational cost of the different steps in more detail.

Figure 6.4 describes different steps of a conventional workflow tailored to compress seismic data, transfer them to a processing center, decompress them, and then apply the wavefield separation and deghosting processing steps. The first step in this workflow is to compress seismic data with a conventional algorithm based on a predefined dictionary such as the Wavelet Packet Transform (WPT) algorithm available in Seismic Unix and introduced in Paper I. The data are then quantized and coded in a similar manner to our method (Stockwell and Cohen, 2008). Such algorithms succeed in providing a compressed format of the recorded data with a size that is more than 10 times smaller than the input, while reaching a high SRR value of 30 dB. However, once the data are transferred to a processing center, they need to be immediately decompressed in order to be processed. This means that the storage advantage is immediately lost as soon as data are to be processed. The decompression is conducted via decoding and WPT inverse transform to recover the original vertical velocity and pressure data in the time domain. Later, these data are split into high and low frequency components, where the high frequencies need to be interpolated before applying the conventional dual-sensor wavefield separation process introduced in Chapter 4 and Paper II. Interpolating the data leads to increasing their size by at least a factor of 2. The low frequencies are also deghosted using a conventional processing-based method (Chapter 4 and Paper III). In both cases, optimized industry-standard FK methods are used (the same methods that were introduced in Papers II and III) The high frequency and low frequency components are later summed to reconstruct the up- and down-going parts of the data.

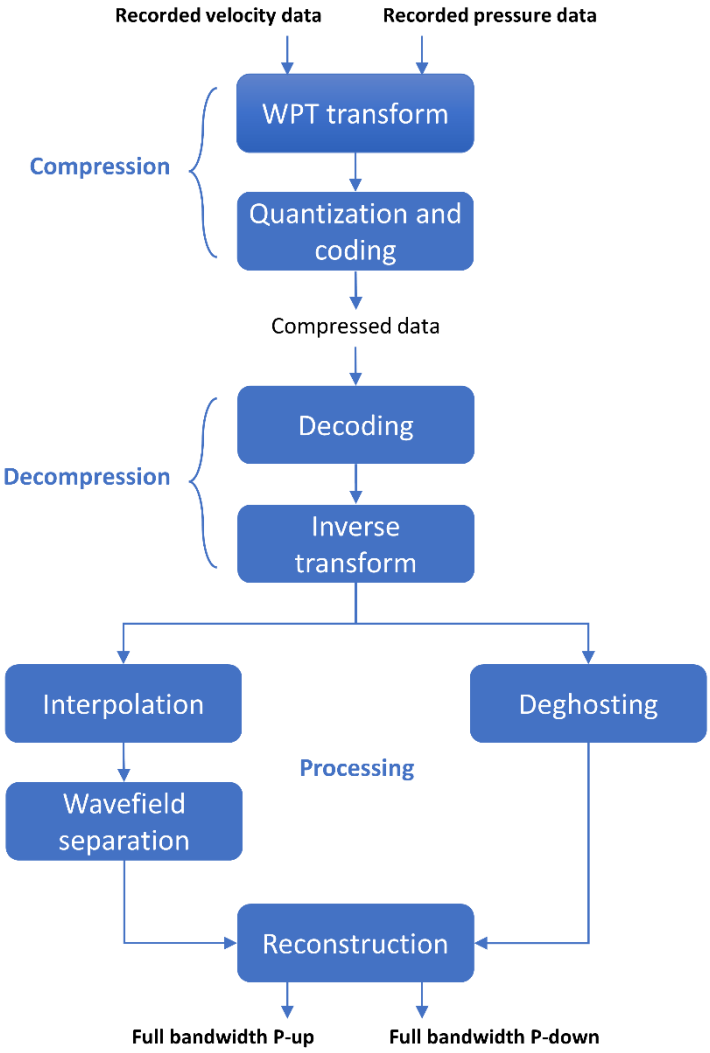

Figure 6.4: *Illustration of a conventional workflow where recorded data are compressed, decompressed, and processed. Here, the processing tasks of interest are dual-sensor wavefield separation and low frequency receiver deghosting.*

We are now in the position to discuss the computational cost of both workflows. Both workflows were run on a computer with an Intel Xeon processor running at 2.93 GHz using 60 GB of RAM. Note also that our workflow is a prototype mainly written in MATLAB, while the conventional workflow is an optimized algorithm written in C/C++. Learning the dictionaries $\mathbf{D}_P$, $\mathbf{D}_{V_Z}$, and $\mathbf{D}$ takes 17 min 14 sec, 18 min 56 sec, and 10 min 16 sec, respectively. This means that the whole learning part takes 46 min 26 sec. Here, we notice that learning the dictionary $\mathbf{D}$ takes less time than the two other dictionaries. That is because the learned atoms contain a smaller number of samples given that $\mathbf{D}$ is learned on data sampled at 16 ms in time. However, in our workflow (as in Papers II and III) we learn dictionaries from $M = 10.000$ training patches but apply an extra OMP step with patches extracted from the whole data set. For the high frequency component data, $L = 678.300$ patches are extracted, and for the low frequency component data, $L = 324.500$ patches are extracted. Hence, it takes 24 min 9 sec and 29 min 45 sec to find the sparse representations corresponding to P_H and V_Z, respectively while it takes only 2 min 59 sec to find the sparse representation corresponding to P_L. This is due to the small number of extracted patches, the small number of samples in each patch, but also OMP needs fewer iterations to reach the user defined threshold (Rubinstein, 2008) when applied to P_L. Hence, it takes 102 min and 53 sec in total to go from the time domain to the DL compressed domain for the 100 shots synthetic data set with our workflow. That includes both the learning time and the OMP time to find the sparse representations. Then, applying a scaling operator to each trace of $\mathbf{D}_{V_Z}$ atoms is a fast process taking only 1 sec, tapering and deghosting $\mathbf{D}$ atoms takes 1 min 48 sec. The quantization and coding stages took around 15 min 49 sec (of which less than 30 sec is used for quantization), for all the dictionaries $\mathbf{D}_{CV_Z}$, $\mathbf{D}_P$, and $\mathbf{D}_u$, and their corresponding sparse representations. In contrast, the WPT-based algorithm takes around 1 min 32 sec to compress, quantize and code both data sets. Given that the WPT-based algorithm uses a similar Huffman coding (Nelson and Gailly, 1995) to the one we use in our workflow (Skretting et al., 1999) as mentioned in Paper I, it would be unfair to directly compare the run time of both coding strategies since the computation cost difference depends mainly on the implementation of the Huffman scheme. Hence, we consider the run time of compression, coding, and quantization to be negligible when the WPT-based algorithm is used, and we do not consider the quantization and coding run times in our workflow. The same applies to the decoding that takes around 1 min 2 sec with the conventional workflow, and which depends only on the Huffman coding implementation as well (Paper I). After decoding and applying the inverse of the WPT transform, the wavefield separation is applied in the conventional workflow case (Figure 6.4). It takes 114 min 26 sec to obtain P_H^{up} and P_H^{down}. Note that more than 75% of this time is

spent to interpolate the data. Then, to deghost the low-frequency component, it takes around 2 min 18 sec. Note that this is less than twice the run time to deghost our atoms, while we have shown in Paper III that deghosting the atoms with our workflow can theoretically reduce the computational cost by a factor of 30. This is expected since our method is not optimized as an industry-standard algorithm. Once P_L^{up} is obtained, we sum it with P_H^{up} to obtain P^{up}, the same way it is done with our workflow. Then, to obtain P_L^{down} we simply subtract P_L^{up} from P_L. The computation cost of such reconstruction is negligible since it only involves summations and subtractions after applying the wavefield separation and deghosting processes. In the case of our workflow (Figure 6.3), we should still consider the run time of the different matrix multiplications that allowed us to reconstruct the data back in the time domain. To reconstruct P^{up}, three matrix multiplications occur: two to reconstruct P_H^{up} where $\mathbf{D}_{CV_Z}$, $\mathbf{D}_P$ are multiplied by their sparse representations, and one to reconstruct P_L^{up}. Hence it takes 24 sec to reconstruct P^{up} and approximately the same time to reconstruct P^{down} . Then, to obtain the original recorded pressure P, P_H is first obtained by a simple matrix multiplication including $\mathbf{D}_P$ and the corresponding set of sparse representations. Then, $\mathbf{D}_u$is used to recover $\mathbf{D}$ as described earlier, which includes applying the deghosting operator again. Later, $\mathbf{D}$ is multiplied by the corresponding set of sparse representations to obtain P_L. The extra deghosting operator and the two matrix multiplications take around 2 min 5 sec. The vertical velocity data V_Z can easily be reconstructed using only one matrix multiplication after rescaling the atoms of $\mathbf{D}_{CV_Z}$ to obtain $\mathbf{D}_{V_Z}$ (which takes 1 sec). This matrix multiplication takes around 8 sec. Thus, in order to reconstruct the whole four data sets in the original time-space domain with our workflow it takes around 3 min. Therefore, the resulting time after summing all the steps of our workflow is around 106 min while all the steps of the conventional workflow take around 116 min. Here, we did not consider the coding and decoding run times for the aforementioned reasons. We also did not consider the resampling times since resampling is applied in both algorithms. Moreover, we have considered that the run times of the WPT forward and inverse transforms are negligible. We can conclude by considering that the run times of both workflows are comparable, even though our workflow has not been optimized for computation cost.

Overall, the workflow we have developed which applies wavefield separation and deghosting directly in the DL compressed domain can have many advantages over the conventional workflow. First, it can release the geophysicists from the wavefield separation and deghosting tasks, which can be expensive in terms of time and human resources. Recall that, in this example, the different steps of the workflow take around 2 hours while we have only considered a small data set of 100 shots. It can also save more storage compared to the conventional

workflow. Indeed, even though the WPT-based algorithm might succeed in reaching comparable compression results to our DL based method, the data needs to be in a decompressed format while being processed, i.e., expressed in the time-space domain, which requires more storage during the period where wavefield separation and deghosting are performed. Also, during processing the data are even interpolated which leads to higher storage requirements. Moreover, after wavefield separation and deghosting, only P^{up} is generally needed to perform other processing tasks. Hence, in the conventional case where the WPT-based algorithm is used for compression, both P and V_Z can be compressed for backup, and the processing can be re-applied later to recover the original P^{up}, and P^{down} if needed, or alternatively, the four data sets can be kept in their compressed format as a backup. In contrast, with our workflow, the compressed P and V_Z can be used to directly reconstruct the four data sets at almost no additional cost. In Figure 6.4, we have considered the wavefield separation and deghosting tasks to be performed after data compression, transfer, and decompression. In other words, we have considered these processing tasks to be performed in processing centers to align with the objectives of Chapter 1. Nevertheless, such processing tasks can also be applied on the vessels, then compressed and transferred for further processing. Even then, applying our workflow would still be more advantageous compared to applying the conventional workflow. Indeed, in such cases, the simultaneous compression, wavefield separation, and deghosting with our workflow will take approximately the same time as wavefield separation, deghosting and then compression with the conventional workflow (around 2h for the 100 shots example). However, while the data are being processed on the vessel, more storage will be needed with the conventional workflow since interpolation increases the size of the data by at least a factor of 2. Furthermore, the four data sets will need to be compressed in order to transfer them to the processing centers instead of only two when our workflow is used. Finally, applying the processing steps in the compressed domain comes almost for free as we could see with wavefield separation and deghosting run times, it is the compression that takes most of the time. Hence, since the compression occurs only once, enabling further processing steps with our workflow would save more costs related to time and storage.

6.2. Outlook

This thesis focused on enabling wavefield separation and receiver deghosting directly in the DL compressed domain only for the 2D case. Future work can focus on generalizing our methods to the 3D case. In the discussion section of Paper II, suggestions were made to achieve such goal with the dual-sensor wavefield separation process. The same suggestions can be applied to the low-frequency deghosting process (Paper III). Indeed, both processes need information related to kinematic parameters (slope and curvature) in the crossline direction to enable this generalization. For instance, kinematic parameters in the crossline direction can be estimated by learning 3D atoms characterized by five parameters (two slopes, two curvatures, and one parameter correlating both directions) instead of only two parameters in the 2D case. In the Common Reflection Surface (CRS) theory, these parameters have been estimated to achieve 3D interpolation (Hoecht et al., 2019). However, such an approach might be computationally expensive. Another way to estimate these parameters is to consider a pseudo-3D approach. For instance, Klüver and Day (2011) suggested a method in which each streamer is processed several times using 2D plane-wave decomposition, with a different constant crossline slowness assumed each time. This approach enabled the processing of 3D dual-sensor towed streamer data without requiring proper 3D plane wave decomposition.

In Papers II and III we have used PDL (Turquais et al., 2018) to compress seismic data. PDL characterizes the different atoms with local kinematic parameters. These parameters were used to allow processing but can also be used to improve the compression performance of our methods. Indeed, the PDL approach constrains the atoms by stacking the traces, averaging them, and then shifting them back according to a given parabolic time moveout. Therefore, one can imagine storing only the averaged trace along with its corresponding slope and curvature, instead of storing all coefficients of 2D atoms. This can significantly reduce the dictionary size and thus improve the compression performance of our method. The computational cost of our method may also be reduced using transfer learning. Transfer learning is a popular technique in machine learning that involves using a model trained on one data set to make predictions on a different data set. In the context of the given statement, transfer learning can also help to reduce the learning cost of our method. Specifically, the DL-based methods developed in this thesis can employ transfer learning to learn one dictionary containing various slopes and curvatures on a particular data set and then use that dictionary with different data sets. This approach may help reducing the number of stored dictionaries and thus improve compression performance, while also reducing the overall computational cost.

In this thesis, DL-based methods have been constrained with the physics of the wavefield to both compress and process seismic data. While dictionary learning is a subfield of machine learning, it has its own set of algorithms based on the wide field of sparse approximation. These methods have been shown to be tightly connected to the very popular Neural Networks (NN), which led to state-of-the-art results in various fields. More details regarding the relation between sparse coding and neural networks can be found in Papyan et al., (2017). While this work constrained dictionary atoms with the physics of the wavefield, future research may explore the feasibility of developing similar approaches where NN filters are constrained to learn kinematic parameters of the wavefield.

The research conducted in this thesis has made it possible to separate the wavefield and eliminate the receiver ghosting effects in the compressed domain. This opens up opportunities for further research to focus on additional processing steps in this domain. For instance, it is possible to integrate denoising into the workflow as both coherent and incoherent noise removal processes have already been established in the DL domain (Beckouche and Ma, 2014; Turquais et al., 2017a; Turquais et al., 2017b; Gomez and Velis, 2020). As mentioned in Paper II, dip filtering may also be applied in the DL compressed domain since local velocity information can be accessed from the learned atoms. Paper III also mentions the potential for source deghosting using the common receiver domain. Additionally, demultiple processing has already been performed using curvelet frames (Herrmann et al., 2008). Therefore, future research can focus on primary-multiple separation using PDL. Lastly, future work could concentrate on enabling imaging in the DL compressed domain, as estimating local kinematic attributes from pre-stack data has been shown to facilitate seismic imaging (Fomel, 2007), and sparse transforms have already been used for seismic migration (Chauris and Nguyen, 2008).

Bibliography

Aharon, M., M. Elad, and A. Bruckstein, (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing,* 54, 4311-4322.

Akbari, A., and M. Trocan, (2019). On the application of dictionary learning to image compression. *Coding Theory. IntechOpen.* DOI: 10.5772/intechopen.89114.

Amundsen, L., (1993). Wavenumber-based filtering of marine point-source data. *Geophysics,* 58, 1335-1348.

Amundsen, L., B. Arntsen, A. Reitan, E. Ø. Dischler, and B. Ursin, (2008). Wave equation depth migration — A new method of solution. *78th Annual International Meeting, SEG, Expanded Abstracts*, 2252–2256.

Aytun, K., (1999). The footsteps of the receiver ghost in the FK domain. *Geophysics,* 64, 1618–1626.

Beckouche, S., and J. Ma, (2014). Simultaneous dictionary learning and denoising for seismic data. *Geophysics,* 79, A27–A31.

Bortfeld, R., (1989). Geometrical ray theory: Rays and traveltimes in seismic systems (second-order approximations of the traveltimes). *Geophysics,* 54, 342–349.

Bruckstein, A. M., D. L. Donoho, and M. Elad, (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review,* 51, 34–81.

Candès, E. J., and L. Demanet, (2005). The curvelet representation of wave propagators is optimally sparse. *Communications on Pure and Applied Mathematics*, 58, 1472–1528.

Candès, E. J., and D. L. Donoho, (2000). Curvelets: a surprisingly effective nonadaptive representation of objects with edges. Curve and surface fitting. *Vanderbilt University Press,* 105–120.

Candès, E. J., and D. L. Donoho, (2002). Recovering edges in ill-posed inverse problems: optimality of curvelet frames. *Ann. Statist.*, 30, 784–842.

Cerveny, V., (2001). Seismic ray theory. *Cambridge University Press.*

Day, A., T. Klüver, W. Söllner, H. Tabti, and D. Carlson, (2013). Wavefield-separation methods for dual-sensor towed-streamer data. *Geophysics,* 78, WA55-WA70.

Deutsch, L. P., (1996). DEFLATE compressed data format specification version 1.3. *Internet Engineering Task Force (IETF). Abstract p.1.*

Donoho, D. L., and M. Elad, (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell 1$ minimization. *Proceedings of the National Academy of Sciences of the United States of America,* 100, 2197-2203.

Donoho, D. L., (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52, 1289-1306.

Duval, L. C., and T. Q. Nguyen, (1999). Seismic data compression: A comparative study between GenLOT and wavelet compression. *Proceedings of SPIE*, 3813, 802-810.

Duval, L. C., and T. Rosten, (2000). Filter bank decomposition of seismic data with application to compression and denoising. *70th SEG Annual International Meeting*, *Expanded Abstracts*, 2055–2058.

Elad, M., P. Milanfar, and R. Rubinstein, (2007). Analysis versus synthesis in signal priors. *Inverse problems*, 23, 947.

Elad, M., (2010). Sparse and redundant representations: From theory to applications in signal and image processing. *1st ed.: Springer Publishing Company.*

Fajardo, C., O. M. Reyes and A. Ramirez, (2015). Seismic data compression using 2D lifting-wavelet algorithms. *Ingenieria y Ciencia,* 11, 221-238.

Faouzi Zizi, M. O., and P. Turquais, (2022). A dictionary learning method for seismic data compression. *Geophysics,* 87, V101-V116.

Faouzi Zizi, M. O., P. Turquais, A. Day, M. W. Pedersen, and L. J. Gelius, (2023). Dual-sensor wavefield separation in a compressed domain using parabolic dictionary learning. *Geophysical prospecting*, 1-19.

Fomel, S., (2007). Velocity-independent time-domain seismic imaging using local event slopes. *Geophysics*, 72, S139-S147.

Fomel, S., and Y. Liu, (2010). Seislet transform and seislet frame. *Geophysics*, 75, V25–V38.

Foster, D. J., C. C. Mosher, and S. Hassanzadeh, (1994). Wavelet transform methods for geophysical applications. *SEG Technical Program Expanded Abstracts 1994*, 1465–1468

Ghosh, S. K., (2000). Deconvolving the ghost effect of the water surface in marine seismic. *Geophysics*, 65, 1831-1836.

Gomez, J. L., and D. R. Velis, (2020). Footprint removal from seismic data with residual dictionary learning. *Geophysics*, 85, V355-V365.

Herrmann, F. J., D. Wang, and D. J. Verschuur, (2008). Adaptive curvelet-domain primary-multiple separation. *Geophysics,* 73, A17-A21.

Hennenfent, G., and F. Herrmann, (2006). Seismic denoising with nonuniformly sampled curvelets. *Computing in Science and Engineering, 8, 16-25.*

Hoecht, G., P. Ricarte, S. Bergler, and E. Landa, (2009). Operator-oriented crs interpolation. *Geophysical Prospecting,* 57, 957–979.

Hubral, P., (1983). Computing true amplitude reflections in a laterally inhomogeneous earth: *Geophysics*, 48, 1051–1062.

Hubral, P., J. Schleicher, and M. Tygel, (1992). Three-dimensional paraxial ray properties: Part I. Basic relations. *Journal of Seismic Exploration,* 1, 265-279.

Huffman, D. A., (1952). A method for construction of minimum redundancy codes. *Proceedings of the IRE,* 40, 1098–1101.

Kiselev, A. P., (2007). Localized light waves: Paraxial and exact solutions of the wave equation (a review). *Opt. Spectrosc.* 102, 603–622.

Klüver, T., and A. Day, (2011). Processing 3-D dual-sensor towed streamer data using local crossline slowness estimates. *81st Annual International Meeting, SEG, Expanded Abstracts,* 3556–3560.

Mairal, J., F. Bach, J. Ponce, and G. Sapiro, (2009). Online dictionary learning for sparse coding. *26th Annual International Conference on Machine learning*, 689-696.

Mallat, S., (2008). A wavelet tour of signal processing: *The sparse way. 3rd ed.: Academic Press.*

Masoomzadeh, H., and N. Woodburn, (2013). Broadband processing of conventional streamer data - Optimized de-ghosting in the tau-p domain. *75th Annual International Conference and Exhibition, EAGE, Extended Abstracts.*

Mohanalin, J., S. Prabavathy, J. Torrents-Barrena, and D. Puig, (2016). A Novel Wavelet Seismic Denoising Method Using Type II Fuzzy. *Applied Soft Computing*, 48, 507–521.

Naghizadeh, M., and M. D. Sacchi, (2010a). On sampling functions and Fourier reconstruction methods. *Geophysics*, 75, WB137–WB151.

Naghizadeh, M., and M. D. Sacchi, (2010b). Beyond alias hierarchical scale curvelet interpolation of regularly and irregularly sampled seismic data. *Geophysics*, 75, WB189–WB202.

Neelamani, R., A. I. Baumstein, D. G. Gillard, M. T. Hadidi, and W. L. Sokora, (2008). Coherent and random noise attenuation using the curvelet transform. *The leading Edge*, 27, 240-248.

Nelson, M., and J. L. Gailly, (1995). The data compression book, *2nd ed.: M&T Books.*

Papyan, V., Y. Romano, M. Elad, (2017). Convolutional neural networks analyzed via convolutional sparse coding. *Journal of Machine Learning Research,* 18, 1-52.

Pati, Y. C., R. Rezaiifar, and P.S. Krishnaprasad, (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, 40-44.

Robertsson, J., and L. Amundsen, (2014). Deghosting of arbitrarily depth-varying marine hydrophone streamer data by time-space domain modelling. *SEG expanded abstracts*, 4248–4542.

Robertsson, J., I. Moore, M. Vassallo, K. Özdemir, D.-J. van Manen, and A. Özbek, (2008). On the use of multicomponent streamer recordings for reconstruction of pressure wavefields in the crossline direction. *Geophysics,* 73, A45–A49.

Robinson, E., and S. Treitel, (2008). Digital Imaging and Deconvolution: the ABCs of Seismic Exploration and Processing. *Tulsa: Society of Exploration Geophysicists.*

Rubinstein, R., M. Zibulevsky, and M. Elad, (2008). Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *Technion report* CS-2008-08.

Schonewille, M., A. Klaedtke, and A. Vigner, (2009). Anti-alias anti-leakage Fourier transform. *SEG Technical Program Expanded Abstracts 2009*, 3249–3253.

Skretting, K., J. H. Husœy, and S. O. Aase, (1999). Improved Huffman coding using recursive splitting. *Proceedings of the NORSIG-99 Conference*, 92–95

Skretting, K., and K. Engan, (2011). Image compression using learned dictionaries by RLS-DLA and compared with K-SVD. *IEEE International conference on Acoustics, Speech, and Signal Processing*, pp. 1517-1520.

Smith, W., (1997). The scientist and engineer's guide to digital signal processing. *California Technical Publishing.*

Söllner, W., A. Day, and H. Tabti, (2008). Space frequency domain processing of irregular dual-sensor towed streamer data. *SEG expanded abstracts, 27*, 1078-1083.

Sweldens, W., (1995). Wavelets and the lifting scheme: A 5 minute tour. *Journal of Applied Mathematics and Mechanics*, 76, 41–47.

Tropp, J., (2004). Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50, 2231-2242.

Taubman, D., M. Marcellin, (2002). JPEG2000 Image Compression fundamentals, standards and practice. *The International Series in Engineering and Computer Science,* 800 p.

Turquais, P., E. G. Asgedom, and W. Sollner, (2017a). A method of combining coherence-constrained sparse coding and dictionary learning for denoising. *Geophysics,* 82, V137–V148.

Turquais, P., E. G. Asgedom, and W. Sollner, (2017b). Coherent noise suppression by learning and analyzing the morphology of the data. *Geophysics,* 82, V397–V411.

Turquais, P., E. G. Asgedom, W. Söllner, and L.-J. Gelius, (2018). Parabolic dictionary learning for seismic wavefield reconstruction across the streamers. *Geophysics,* 83, V263-V282.

Ursin, B., (1982). Quadratic wavefront and traveltime approximations in inhomogeneous layered media with curved interfaces. *Geophysics*, 47, 1012–1021.

Vassallo, M., A. Özbek, Y. Kamil, D. van Manen, K. Eggenberger, (2014). Reconstruction of signals from highly aliased multichannel samples by

Generalized Matching Pursuit. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2382-2385

Villasenor, J. D., R. A. Ergas and P. L. Donoho, (1996). Seismic data compression using high-dimensional wavelet transforms. *Proceedings of Data Compression Conference - DCC '96, Snowbird, UT, USA*. pp. 396-405.

Wang, X., Y. Teng, M. Gao, and H. Jiang, (2004). Seismic data compression based on integer wavelet transform: *Acta Seismologica Sinica,* 17**,** 123-128.

Wang, Y., A. C. Ramirez, and S. Nag, (2016). A 3D deghosting solution for pressure-only measurements of conventional towed streamer data: *78th Annual International Conference and Exhibition, EAGE, Extended Abstracts*, We-SRS3-15.

Wang, Y., A. C. Ramirez, and A. Osen, (2017). A low-frequency deghosting method: Analysis and numerical tests. *Geophysics,* 82, V285-V296.

Xu, S., Y. Zhang, D. Pham, and G. Lambaré, (2005). Antileakage Fourier transform for seismic data regularization. *Geophysics*, 70, V87-V95.

Zhang, Y., S. Bergler, and P. Hubral, (2001). Common-reflection-surface (CRS) stack for common offset. *Geophysical Prospecting,* 49, 709–718.

Zhang, Z., H. Masoomzadeh, and B. Wang, (2018). Evolution of deghosting process for single-sensor streamer data from 2D to 3D. *Geophysical Prospecting,* 66, 975-986.

Zhang, Z., Z. Wu, B. Wang, and J. Ji, (2015). Time variant de-ghosting and its applications in WAZ data. *SEG expanded abstracts*, 4600–4604.

Zheng, F., and S. Liu, (2012). A fast compression algorithm for seismic data from non-cable seismographs: *World Congress on Information and Communication Technologies*, 1215–1219.

Zhivomirov, H., (2019). Inverse short-time Fourier transform with MATLAB. *MATLAB Central File Exchange*.
https://www.mathworks.com/matlabcentral/fleexchange/45577-inverse-short-time-fouri er-transform-istft-with-matlab.

Zwartjes, P. M., and M. D. Sacchi, (2007). Fourier reconstruction of nonuniformly sampled, aliased seismic data. *Geophysics*, 72, V21–V32.

Papers

Paper I

A dictionary learning method for seismic data compression

Mohammed Outhmane Faouzi Zizi [1,2], **Pierre Turquais**[1]

Published in *Geophysics*, 2022, volume 87, No 2, P. V101–V116.
DOI: 10.1190/geo2020-0948.1.

[1]PGS Geophysical AS, Oslo, Norway.
[2]University of Oslo, Department of Geosciences, Oslo, Norway.

Paper II

Dual-sensor wavefield separation in a compressed domain using parabolic dictionary learning

Mohammed Outhmane Faouzi Zizi [1,2], **Pierre Turquais**[1], **Anthony Day**[1], **Morten W. Pedersen**[1], **Leiv J. Gelius**[2]

Published in *Geophysical Prospecting*, 2023, DOI: 10.1111/1365-2478.13348.

II

[1]PGS Geophysical AS, Oslo, Norway.
[2]University of Oslo, Department of Geosciences, Oslo, Norway.

Received: 11 August 2022 | Accepted: 28 February 2023

DOI: 10.1111/1365-2478.13348

Geophysical Prospecting EAGE EUROPEAN ASSOCIATION OF GEOSCIENTISTS & ENGINEERS

ORIGINAL ARTICLE

Dual-sensor wavefield separation in a compressed domain using parabolic dictionary learning

Mohammed Outhmane Faouzi Zizi[1,2] | **Pierre Turquais**[1] | **Anthony Day**[1] | **Morten W. Pedersen**[1] | **Leiv J. Gelius**[2]

[1]PGS Geophysical AS, Oslo, Norway

[2]Department of Geosciences, University of Oslo, Oslo, Norway

Correspondence
Mohammed Outhmane Faouzi Zizi, PGS Geophysical AS, Oslo, Norway.
Email: mohammed.faouzi.zizi@pgs.com

Abstract

In the marine seismic industry, the size of the recorded and processed seismic data is continuously increasing and tends to become very large. Hence, applying compression algorithms specifically designed for seismic data at an early stage of the seismic processing sequence helps to save cost on storage and data transfer. Dictionary learning methods have been shown to provide state-of-the-art results for seismic data compression. These methods capture similar events from the seismic data and store them in a dictionary of atoms that can be used to represent the data in a sparse manner. However, as with conventional compression algorithms, these methods still require the data to be decompressed before a processing or imaging step is carried out. Parabolic dictionary learning is a dictionary learning method where the learned atoms follow a parabolic travel time move out and are characterized by kinematic parameters such as the slope and the curvature. In this paper, we present a novel method where such kinematic parameters are used to allow the dual-sensor (or two-components) wavefield separation processing step directly in the dictionary learning compressed domain for 2D seismic data. Based on a synthetic seismic data set, we demonstrate that our method achieves similar results as an industry-standard FK-based method for wavefield separation, with the advantage of being robust to spatial aliasing without the need for data preconditioning such as interpolation and reaching a compression rate around 13. Using a field data set of marine seismic acquisition, we observe insignificant differences on a 2D stacked seismic section between the two methods, whereas reaching a compression ratio higher than 15 when our method is used. Such a method could allow full bandwidth data transfer from vessels to onshore processing centres, where the compressed data could be used to reconstruct not only the recorded data sets, but also the up- and down-going parts of the wavefield.

KEYWORDS

compression, data processing, dictionary learning, multicomponent, seismics

INTRODUCTION

During the last decade, marine seismic acquisition capabilities have improved at a rapid pace. As a result, the size of marine seismic exploration surveys, and the number of sensors deployed have both increased. Hence, the size of the seismic data recorded during one marine seismic survey has increased significantly and often reaches several terabytes. Such large data sizes give rise to several challenges related to data transfer, storage and processing. For example: Only bandlimited data can typically be transferred from vessels to onshore processing centres because of the low-bandwidth

available from satellites; storing many large seismic data sets for a long time on tapes is costly but necessary as many of them need to be backed up even after project delivery. Hence, compressing data at an early stage of the seismic processing sequence has attracted growing interest from the marine seismic community in the last few years. Indeed, designing efficient compression algorithms for seismic data is a key element to removing the barriers related to storage and data transfer. However, conventional compression processes still require the data to be decompressed before carrying out any processing or imaging steps, by transforming them back into their original time-space domain. This can limit the compression benefits related to storage or lead to accumulating compression errors if the compression process is again applied after carrying out that processing or imaging step. Moreover, conventional seismic processing and imaging is a long meticulous sequence of workflows, which generally requires transforming the data into other processing domains where data preconditioning is needed (e.g. zero padding, data interpolation, data extrapolation and decimation) and comes at a significant computational cost. In this paper, we aim to enable one seismic processing step directly in the compressed domain, namely the dual-sensor (or two-components [2C]) wavefield separation processing step. This would allow to overcome the requirements for data decompression and avoid data preconditioning with regards to this specific processing step, which is generally applied early in the seismic processing sequence.

Many compression algorithms have been designed for seismic data. Such algorithms are generally based on transforming the data to a so-called sparse domain, which is more compact than the original time-space domain. Sparse domains have been used to carry out different seismic processing and imaging applications (compressed sensing). Indeed, the compressed sensing fields have helped to tackle many difficulties related to seismic data starting from acquisition to full waveform inversion by exploiting the sparse structure of seismic data (Herrmann et al., 2013; Lin & Herrmann, 2013; Mansour et al., 2012). Conventionally, seismic compression algorithms are based on fixed sparse transforms (Averbuch et al., 2001; Duval & Rosten, 2000; Fajardo et al., 2015; Wang et al., 2004; Zheng & Liu, 2012), where the basis functions are analytically predefined and already known by the encoder and decoder, such as discrete cosines, wavelets and others (Elad, 2010; Mallat, 2008). By contrast, other seismic compression algorithms based on learned transforms have recently emerged. Schiavon et al. (2020) proposed a deep autoencoder to compress post-stack seismic data. Helal et al. (2021) proposed two convolutional autoencoders, where the first model is adapted to low compression rates (CRs), whereas the second model is more efficient when the user needs to reach high CR. These methods transform the input seismic data into feature representations which are sparse enough to allow good compression performance. Dictionary learning (DL) methods, for example *K*-mean singular value decomposition (Aharon et al., 2006) or online DL (Mairal et al., 2009), are another type of learned transform. These methods capture the similar elementary events from the seismic data, store them once in a dictionary of atoms and then express the original data as a weighted sum of the learned atoms. DL methods have been shown to provide state-of-the-art results when it comes to seismic data compression (Faouzi Zizi & Turquais, 2022). The authors have developed a compression workflow where the similarities between the different seismic events is fully exploited, and where the DL-based compression algorithm provides better compression performance compared to conventional compression methods. Moreover, different modifications of the DL methods have been shown to be suited to various seismic processing tasks, such as noise suppression (Beckouche & Ma, 2014) or interpolation (Turquais et al., 2018). For example Turquais et al. (2018) proposed a parabolic dictionary learning (PDL) method where the learned atoms represent elementary waveforms of constant amplitude along parabolic travel time move out. Hence, each atom can be characterized by a set of parameters such as the slope and the curvature, which relate to the kinematics of the wavefield (Bortfeld, 1989; Ursin, 1982). These kinematic parameters are then used to interpolate the atoms along their respective slopes, thereby reconstructing the interpolated single-component 3D streamer data in the crossline direction. These local kinematic parameters can be used not only for interpolation but also for other processing tasks such as dual-sensor wavefield separation.

In dual-sensor towed streamer acquisition, also referred as 2C streamer data, wavefield separation is the process of decomposing the data into upward and downward travelling waves using two types of sensors: hydrophones and geophones. The hydrophones record the pressure, and the geophones record the vertical component of particle velocity at the same locations. Combining both records facilitates removal of the receiver-side sea surface ghost reflection to produce data with better resolution than data with the ghost present (Söllner et al., 2008). However, for emergence angles greater than zero, the amplitudes recorded by the geophones need to be scaled by an obliquity correction factor as only the vertical component of the particle velocity is recorded (Amundsen, 1993; Söllner et al., 2008).

In this work, we use PDL to both compress seismic data and extract the kinematic parameters from parabolic atoms such as the slope and the curvature. These kinematic parameters are further used to derive the obliquity correction factor for local events in the time domain which allows the dual-sensor wavefield separation processing step to be carried out directly in the DL compressed domain. Our PDL method for wavefield separation (WSPDL) is benchmarked against an optimized industry-standard FK method for wavefield sepa-

ration (FK-WS) using a synthetic data set. Later, a field data set comprising a full 2D sail-line of marine seismic acquisition is used to assess the differences between both methods. Hence, the WSPDL method shows its robustness with respect to spatial aliasing without the need for data interpolation as for FK-WS methods. Finally, the method also succeeds in reaching high compression levels, where the compressed data can be used to reconstruct not only the recorded data sets, but also the up- and down-going parts of the wavefield.

METHODOLOGY

In order to understand how the PDL method for wavefield separation (WSPDL) enables dual-sensor wavefield separation in the compressed domain, it is first necessary to introduce the conventional dictionary learning (DL) and parabolic dictionary learning (PDL) problems.

Conventional dictionary learning and parabolic dictionary learning problems

In conventional DL problems, the aim is to represent the original data in a sparse manner with two parts: A dictionary of learned atoms representing elementary waveforms that are repeated many times in the data; and a set of sparse coefficient vectors. In the case of seismic data, M small 2D patches of size N are first extracted and then vectorized to construct a training set denoted $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M$ (Elad, 2010), which is a subset of the original data set. Then, a DL method such as the K-SVD (K-times the singular value decomposition) algorithm (Rubinstein et al., 2008) is applied to jointly: learn a redundant dictionary $\mathbf{D} \in \mathrm{R}^{N\times K}$ where $N < K$ (Donoho & Elad, 2003) of K atoms, each of size N, same size as the patches, denoted $[\mathbf{d}_1, \mathbf{d}_2 \dots \mathbf{d}_K]$, with $K < M$; and find the set of sparse coefficient vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M$ that minimize the representation error given a sparsity error ϵ imposed on the sparse coefficient vectors (Aharon et al., 2006). This approach is generally referred to as the *error constraint mode*, and this problem can be mathematically expressed by

$$\min_{\{\mathbf{x}_i\}_{i=1}^{M},\mathbf{D}} \sum_{i=1}^{M} \| \mathbf{x}_i\|_0 \text{ subject to } \| \mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 \le \epsilon, \quad i = 1, \dots M. \tag{1}$$

After learning the dictionary $\mathbf{D}$, the sparse optimization problem can be solved for patches of the original data set (Bruckstein et al., 2009). This problem is mathematically expressed as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \| \mathbf{x}\|_0 \text{ subject to } \| \mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \le \epsilon. \tag{2}$$

Equation (2) consists of finding the vector $\mathbf{x}$ of sparse coefficients that minimize the norm of the residual vector for a given patch $\mathbf{y}$ of the original data given the dictionary $\mathbf{D}$, where a sparsity error ϵ is tolerated. We can solve the problem by finding an approximate solution $\hat{\mathbf{x}}$ using orthogonal matching pursuit (Pati et al., 1993). Each patch $\mathbf{y}$ of the original data set can now be described as a linear combination of the dictionary atoms.

The PDL (Turquais et al., 2018) is a modification of the conventional DL method where a geometrical structure is imposed to the atoms while learning them. The parabolic structure was used by Turquais et al. (2018) for interpolation purposes, because it is consistent with the physics of wavefield propagation (Hoecht et al., 2009; Hubral et al., 1992; Zhang et al., 2001). The PDL problem may be mathematically expressed as follows:

$$\min_{\{\mathbf{x}_i\}_{i=1}^{M},\{\mathbf{d}_k\}_{k=1}^{K}} \sum_{i=1}^{M} \| \mathbf{x}_i\|_0 \text{ subject to } \begin{cases} \| \mathbf{y}_i - \sum_{k=1}^{K} \mathbf{d}_k\mathbf{x}_i\|_2 \le \epsilon, \ i = 1, \dots, M \\ \mathbf{d}_k\left(t, o_k^{\mathrm{ref}}\right) = \mathbf{d}_k\left(t + c_k\Delta\mathbf{o}^2 + s_k\Delta\mathbf{o},\ o_k^{\mathrm{ref}} + \Delta\mathbf{o}\right), \ \forall (t, \Delta\mathbf{o}) \in \mathrm{R}^2. \end{cases} \tag{3}$$

Equation (3) is similar to Equation (1) with an extra constraint imposed on the geometrical structure of the atoms. Here, each learned atom $\mathbf{d}_k$ is characterized by a parabolic travel time move out given by $\Delta\mathbf{t} = s_k\,\Delta\mathbf{o} + c_k\Delta\mathbf{o}^2$, where $\Delta\mathbf{t}$ is the time move out, s_k the atom's slope, c_k the atom's curvature and $\Delta\mathbf{o} = [o_k^1 - o_k^{\mathrm{ref}}, o_k^2 - o_k^{\mathrm{ref}}, \dots, o_k^n - o_k^{\mathrm{ref}}]$ the vector containing the displacement of a receiver location o_k^i related to each trace i of an atom k containing n traces relative to the reference receiver o_k^{ref}.

A more detailed description of the PDL problem and the method to find an approximate solution to it can be found in Turquais et al. (2018). Note that in this paper the PDL method is used with the *error constraint mode*, whereas in Turquais et al. (2018) the PDL is used with the *sparsity constraint mode*. That is because the *error constraint mode* is more appropriate when the compressed seismic data are further processed, whereas the *sparsity constraint mode* is more appropriate for visualization purposes as explained in detail by Faouzi Zizi and Turquais (2022).

The WSPDL method

Now, we will describe and illustrate with examples the different stages of the PDL method for wavefield separation (WSPDL), where the dual-sensor wavefield separation process is applied in the DL compressed domain. First, we show how the transformation to the PDL domain is carried out. Then, we derive the obliquity correction factor for

each trace of the velocity dictionary atoms. Next, we combine the corrected velocity and pressure dictionaries to obtain the up-going pressure dictionary and do the same with the velocity and pressure sparse representations to obtain the up-going pressure sparse representation. Further, compression techniques called entropy coding are used in the DL domain to encode the data with a low number of bits. Finally, we reconstruct the up-going pressure data.

Transformation stage

To describe the transformation stage of the WSPDL method, we need to consider a data set example of many shots where both the total vertical components of particle velocity and the total pressure wavefield were recorded by the geophones and the hydrophones, respectively. For reasons of simplicity, we will refer to these data sets as velocity and pressure data sets and will denote them with the indices V_z and P, respectively. The extraction and PDL stages are illustrated in Figure 1. First, a large number M of non-overlapping time-space 2D patches of size N are randomly extracted from both seismic data sets, and vectorized to obtain the training sets denoted $\mathbf{Y}_{V_z}$ and $\mathbf{Y}_P$. Then, for each training set PDL is applied as described earlier to obtain the two parabolic dictionaries $\mathbf{D}_{V_z} = [\mathbf{d}_1^{V_z}, \ldots, \mathbf{d}_K^{V_z}]$ and $\mathbf{D}_P = [\mathbf{d}_1^{P}, \ldots, \mathbf{d}_K^{P}]$ that optimally represent the training data sets in a sparse manner, where K is the number of learned atoms per dictionary. In this figure, we show only few atoms of both dictionaries. Each atom $\mathbf{d}_K$ is characterized by a set of local kinematic parameters o_k^{ref}, s_k and c_k. Further, both the velocity and pressure data sets are split into L number of 2D overlapping patches of size N as shown in Figure 2. The sparse optimization problem represented in Equation (2) can now be solved for the L patches of each of the velocity and the pressure data sets to obtain the sparse representations $\mathbf{X}_{V_z} = [\mathbf{x}_1^{V_z}, \ldots, \mathbf{x}_L^{V_z}]$ and $\mathbf{X}_P = [\mathbf{x}_1^{P}, \ldots, \mathbf{x}_L^{P}]$, given a sparsity error ϵ. Figure 2 represents the relationship between the time-space domain and the DL compressed domain. For example multiplying the sparse representation $\mathbf{x}_3^{V_z}$ (yellow frame) denoted 'Sparse 3' in the figure and the dictionary $\mathbf{D}_{V_z}$ will result in the reconstruction of the velocity patch number 3 represented with the yellow frame in the time-space domain. The atom frames in yellow, red and blue in this figure correspond to the atoms used in the different linear combinations to reconstruct the patches 3, 7065 and L, respectively. The different small squares in the sparse representations (represented with the grey scale) denote different values of non-zero coefficients used in the linear combinations. Note that the dictionary and the set of sparse representations are different for the velocity and pressure data sets.

The sparsity error ϵ is generated from a desired level of signal-to-residual ratio (SRR) of 30 dB following Equation (4), as described by Faouzi Zizi and Turquais (2022)

$$\epsilon \approx 10^{-\left(\frac{\text{SRR}}{20}\right)} \times \|\mathbf{d}_{\text{orig}}\|_2 \times \sqrt{\frac{1}{L}}, \tag{4}$$

where

$$\text{SRR} = 20\log_{10}\frac{\|\mathbf{d}_{\text{orig}}\|_2}{\|\mathbf{d}_{\text{orig}}\| - \|\mathbf{d}_{\text{rec}}\|_2}\ (\text{dB}), \tag{5}$$

with $\mathbf{d}_{\text{orig}}$ and $\mathbf{d}_{\text{rec}}$ denoting original and reconstructed data, respectively. The SRR presented in Equation (5) is a metric defining the level of desired signal distortion on the reconstructed signal. A high level of SRR = 30 dB means that a very small energy loss of 0.1% is expected between the original velocity and pressure data sets and the reconstructed ones after multiplying the dictionaries $\mathbf{D}_{V_z}$ and $\mathbf{D}_P$ with their corresponding set of sparse coefficient vectors $\mathbf{X}_{V_z}$ and $\mathbf{X}_P$. However, higher SRR leads to lower compression rate (CR), which is defined as

$$\text{CR} = \frac{\text{number of bits before compression}}{\text{number of bits after compression}}. \tag{6}$$

Explanations regarding how the data are compressed come in a later section. We will now discuss the choices for the different parameters specified in the WSPDL method based on the 100 shots data set example illustrated in Figures 1 and 2. The data are sampled at 2 ms in time and 12.5 m in space. The number of time samples (Nt) and the number of channels (Nx) are $Nt = 1850$ and $Nx = 475$, respectively. M, the number of extracted patches used to learn the dictionary, was set to 10,000 because that was found sufficient to capture most of the similar features in a data set comprising 100 shots. Although the number of atoms is generally set to be at least five times smaller than the number of training patches in conventional DL methods, here we have set K, the number of atoms per dictionary, to 6000 to enforce redundancy in the dictionary to ensure that a high level of sparsity is reached. The greater the redundancy in a dictionary, the sparser and more accurate will be the representations (Donoho, 2006). However, increasing the size of the dictionary makes the learning stage computationally more expensive. The size of the atoms was set to 64 samples by 8 traces because it was large enough to capture the parabolic move out given the data sampling of 2 ms and 12.5 m. Note that the display of the patch size was increased in Figures 1 and 2 for better visualization. Moreover, the PDL was slightly modified here to better suit our wavefield separation application. An additional constraint was imposed where only atoms with a slope value lower than 1/1500 s/m were learned, as the apparent velocity cannot be smaller than the water velocity. This was done to avoid learning non-physical events, given that the slopes of the different

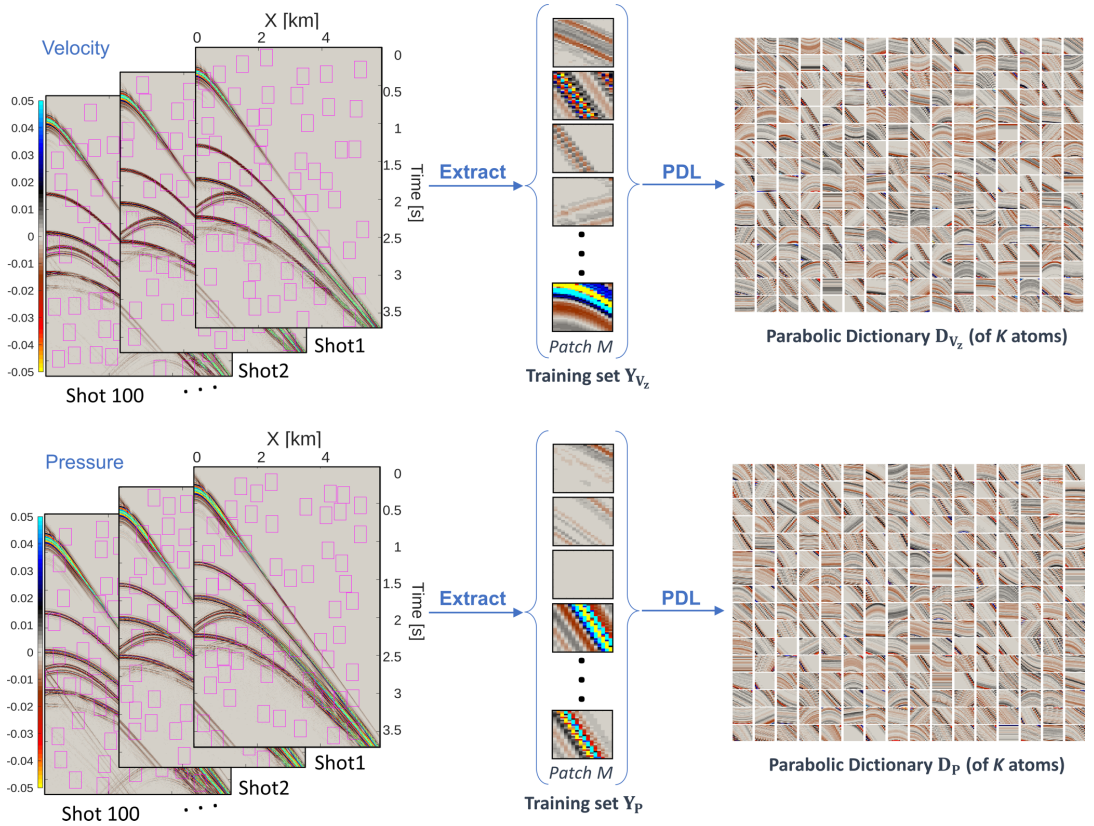

FIGURE 1 Random extraction of M non-overlapping patches from the original data sets recorded by the geophones and the hydrophones to construct the training sets $\mathbf{Y}_{V_z}$ and $\mathbf{Y}_P$, respectively. Then, parabolic dictionary learning (PDL) is applied to obtain the corresponding parabolic dictionaries $\mathbf{D}_{V_z}$ and $\mathbf{D}_P$. Only the first 256 atoms of each dictionary are represented here. The display size of the non-overlapping patches was increased for better visualization.

atoms will later be used to correct for the obliquity scaling problem in the velocity data.

Obliquity correction

The amplitudes recorded by the geophones need to be scaled by an obliquity correction factor F, which can be expressed as follows in the following equation:

$$F = \frac{1}{\cos\theta}, \tag{7}$$

where θ is the emergence angle of each single event. This scaling is needed because the geophones record only the vertical component of particle velocity in dual-sensor acquisition. Such approach assumes that the cable is horizontal, and the propagation velocity is constant at the cable depth (Söllner et al., 2008). In the time domain, the equations for the up- and down-going pressure fields (P^{up} and P^{down}, respectively) can be written as in the following equations:

$$P^{\text{up}} = \frac{1}{2}\left(P - \rho v F V_z\right), \tag{8}$$

$$P^{\text{down}} = \frac{1}{2}\left(P + \rho v F V_z\right), \tag{9}$$

where ρ is the water density, v is the propagation velocity in water, P is the recorded pressure and V_z is the recorded vertical particle velocity. However, it is difficult to find the obliquity correction factor F for every single event in the time domain. Hence, it is more convenient to apply the obliquity scaling after plane wave decomposition, which requires preconditioning (e.g. data interpolation, zero padding) and comes at significant computational cost. The f–k domain is

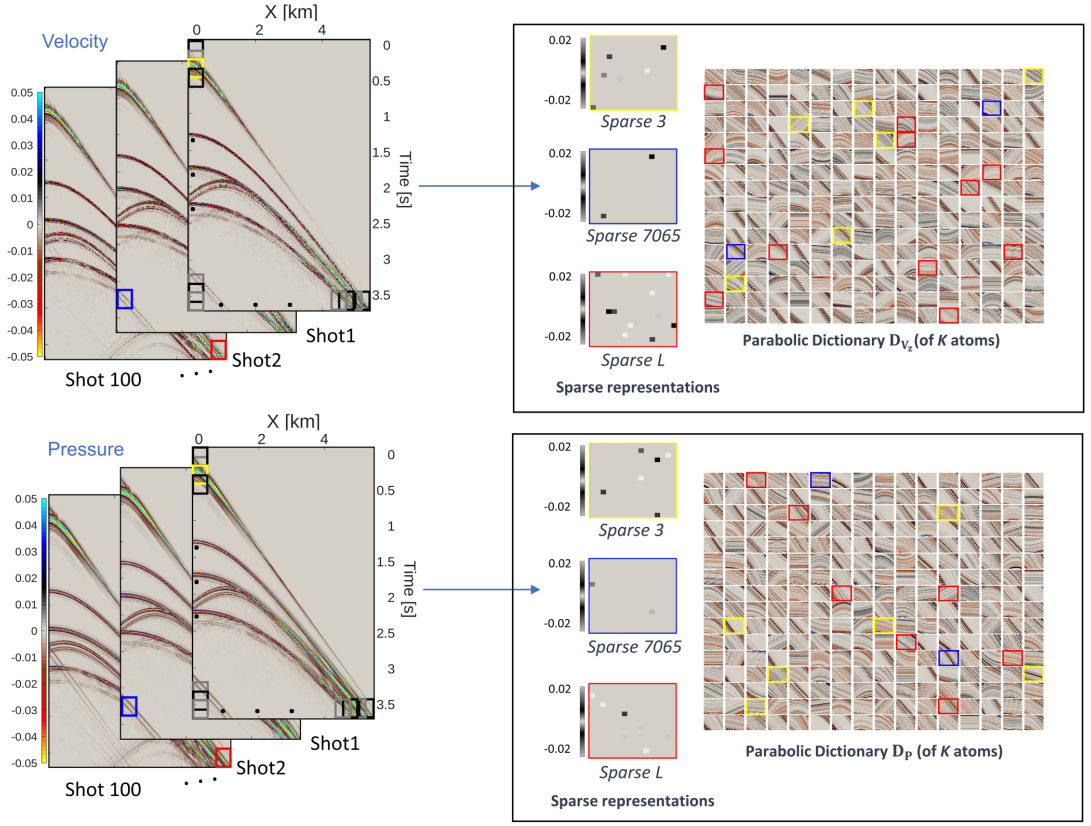

FIGURE 2 Illustration of the data sets recorded by the geophones and the hydrophones in the compressed dictionary learning (DL) domain. In this domain, each seismic data set is represented by a parabolic dictionary and a set of sparse representations. For each data set, three different sparse representations and their corresponding atoms are represented with frames of the same colour (yellow, blue and red). The display of the patch size was increased for better visualization.

conventionally used because each plane wave component maps to a particular coefficient in f–k space (Day et al., 2013). In the f–k domain, Equations (8) and (9) can be rewritten as follows in the following equations:

$$P^{\text{up}}\left(\omega, k_x\right) = \frac{1}{2}\left(P\left(\omega, k_x\right) - \frac{\rho\omega}{k_z} V_z\left(\omega, k_x\right)\right), \quad (10)$$

$$P^{\text{down}}\left(\omega, k_x\right) = \frac{1}{2}\left(P\left(\omega, k_x\right) + \frac{\rho\omega}{k_z} V_z\left(\omega, k_x\right)\right), \quad (11)$$

where ω is the angular frequency, k_x is the horizontal angular wavenumber and k_z is the magnitude of the vertical angular wavenumber. In the WSPDL method, we use the local kinematic parameters to correct for the obliquity scaling problem in the velocity data set. Bortfeld (1989), Hubral et al. (1992) and Zhang et al. (2001) relate the parameters of the parabolic move out to the kinematics of the wavefield based on ray theory assumptions. Ray theory assumes the travelling event to be in the vicinity of a central ray that travelled in the subsurface, and smoothly changing amplitudes in the earth model (Bortfeld, 1989; Ursin, 1982). For example in the common shot domain, the slope s_k of an atom k can be linked to the emergence angle θ_k of the trace at the reference position o_k^{ref} as follows:

$$s_k = \frac{\sin\theta_k}{v}. \quad (12)$$

From Equations (7) and (12), we can write:

$$F_k = \frac{1}{\cos\theta_k} = \frac{1}{\sqrt{1 - \left(vs_k\right)^2}}. \quad (13)$$

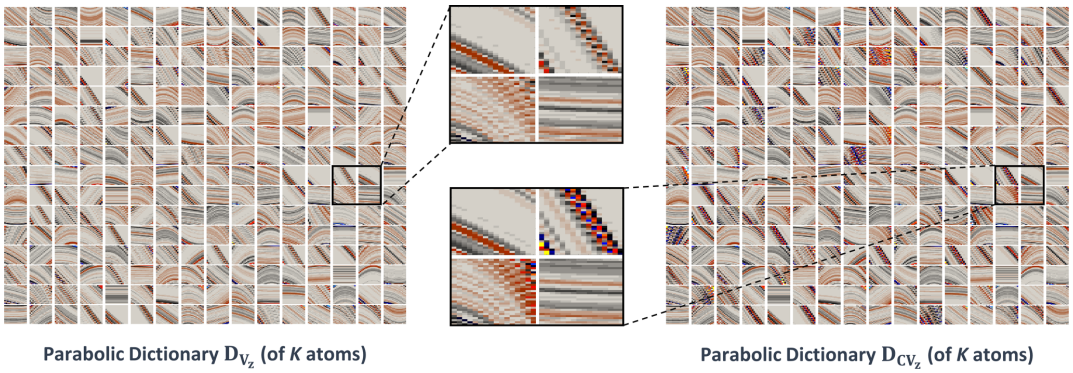

FIGURE 3 Illustration of the velocity dictionary atoms before (left) and after (right) the obliquity correction step, with close-up displays of four different atoms.

In Equation (13), we have found the scaling factor F_k that must be applied to each atom k of the velocity dictionary $\mathbf{D}_{V_z}$ to correct for obliquity scaling. However, the F_k factor we have found here is in fact related only to the trace located at the reference position o_k^{ref} of an atom k. Given that the atoms of our velocity dictionary contain several traces, we need to find the obliquity correction factor F_k^i related to each trace i located at each of the positions o_k^i and not only to the reference position o_k^{ref}. To do so, we consider the derivative of the parabolic time move out function $\Delta\mathbf{t} = s_k\,\Delta\mathbf{o} + c_k\Delta\mathbf{o}^2$. The obliquity correction factor for each trace i at a receiver location o_k^i in an atom k can thus be written as

$$F_k^i = \frac{1}{\sqrt{1-\left(vs_k^i\right)^2}}, \text{ where } s_k^i = 2c_k\,\left(o_k^i - o_k^{\text{ref}}\right) + s_k. \tag{14}$$

Figure 3 shows the velocity dictionary $\mathbf{D}_{V_z}$ on the left side as represented earlier in Figure 2. The atoms of this dictionary are characterized by a slope s_k and a curvature c_k. Hence, we can apply the obliquity correction factors F_k^i to each trace of each atom in the dictionary $\mathbf{D}_{V_z}$ following Equation (14) to obtain the corrected velocity dictionary $\mathbf{D}_{CV_z}$ represented on the right-hand side of Figure 3. The figure also shows the detail of four different atoms before and after applying the obliquity correction. These four atoms displays show that for each single trace, the higher the slope, the higher the angle of incidence and thus the higher the obliquity correction factor F_k^i, which corresponds to the dependency relationship among the obliquity correction factor, the slope and the angle of incidence described in Equations (12)–(14). Moreover, Equation (14) shows that when dealing with high slopes values, small errors in the approximation of the slope might lead to big errors in the estimation of the obliquity correction factor.

Note that both dictionaries have the same set of sparse representations. Multiplying these sparse representations by the dictionary $\mathbf{D}_{CV_z}$ gives the reconstruction of the velocity data set after correcting for obliquity scaling.

Wavefield separation

After scaling the atoms of the velocity dictionary by the different obliquity correction factors following Equation (14), we end up with two parabolic dictionaries: the corrected velocity and the pressure dictionaries denoted $\mathbf{D}_{CV_z}$ and $\mathbf{D}_P$, respectively; and two sets of sparse representations denoted $\mathbf{X}_{V_z}$ and $\mathbf{X}_P$. To allow the dual-sensor wavefield separation process in the DL compressed domain, we need to express the up-going pressure field data set in terms of one dictionary and one set of sparse representations. Thus, we combine the two dictionaries $\mathbf{D}_{CV_z}$ and $\mathbf{D}_P$ into a common up-going pressure field dictionary of $2K$ atoms denoted $\mathbf{D}_{\mathbf{Pup}} \in \mathrm{R}^{N\times 2K}$ such that

$$\mathbf{D}_{\mathbf{Pup}} = \left[\mathbf{D}_{CV_z}\ \mathbf{D}_P\right] = \left[\mathbf{d}_1^{CV_z},\ \ldots,\ \mathbf{d}_K^{CV_z}, \mathbf{d}_1^P,\ \ldots,\ \mathbf{d}_K^P\right], \tag{15}$$

where the first K atoms belong to the corrected velocity dictionary and the last K atoms belong to the pressure dictionary. We also combine the two sets of sparse representations $\mathbf{X}_{V_z} \in \mathrm{R}^{K\times L}$ and $\mathbf{X}_P \in \mathrm{R}^{K\times L}$ into a common up-going pressure field set of sparse representations denoted $\mathbf{X}_{\mathbf{Pup}} \in \mathrm{R}^{2K\times L}$ such that

$$\mathbf{X}_{\mathbf{Pup}} = \begin{bmatrix} -\frac{1}{2}\times\mathbf{X}_{Vz} \\ \frac{1}{2}\times\mathbf{X}_P \end{bmatrix} = \left[\mathbf{x}_1^{\mathbf{Pup}},\ \ldots,\ \mathbf{x}_L^{\mathbf{Pup}}\right], \tag{16}$$

where each sparse representation $\mathbf{x}_i^{\mathbf{Pup}}$ is a sparse vector containing $2K$ coefficients. The rescaling of $\mathbf{X}_{Vz}$ by $\frac{-1}{2}$ and $\mathbf{X}_P$

by $\frac{1}{2}$ is necessary following Equation (8). Now, each patch of the up-going pressure field data set can be described as a linear combination of the dictionary atoms stored in $\mathbf{D_{Pup}}$, where coefficients of each linear combination are stored in $\mathbf{X_{Pup}}$. Hence, one can stay in the DL compressed domain and further processing steps (if enabled) can directly be applied on $\mathbf{X_{Pup}}$ and $\mathbf{D_{Pup}}$ before having to decompress or reconstruct the data.

Data compression

Even though we are already in the DL compressed domain, we further apply mathematical compression techniques commonly known as entropy coding techniques to $\mathbf{D_{Pup}}$ and $\mathbf{X_{Pup}}$ as described in detail by Faouzi Zizi and Turquais (2022). The entropy techniques are applied in two main stages: the quantization and coding stages. The quantization stage is the process of mapping floating point sparse coefficients within a range $[a, b]$ into a finite set of output levels; that is the discretization of a continuous amplitude scale. We adopt the same approach as Faouzi Zizi and Turquais (2022) and rescale the dynamic range of the sparse coefficients by multiplying them by a defined scaling coefficient Sc, which is automatically generated from the level of desired SRR and the norm of the original data $\mathbf{d}_{\text{orig}_2}$. Sc can be mathematically expressed as follows:

$$\text{Sc} = \frac{0.25 \times \sqrt{Nb}}{0.2 \times 10^{-\left(\frac{\text{SRR}}{20}\right)} \times \left|\left|\mathbf{d}_{\text{orig}}\right|\right|_2}, \tag{17}$$

where Nb refers to the total number of non-zero coefficients in the sparse representations. In our method we do the same for the set of sparse representations $\mathbf{X_{Pup}}$. A fixed rescaling factor $\text{Sc} = 10^3$ is also applied to all atoms of the dictionary $\mathbf{D_{Pup}}$ because the dictionary atoms are normalized and do not depend on the scale of the data. Then, we apply a rounding to obtain integer values that can be coded using a small number of bits. For the coding stage, the set of sparse representations $\mathbf{X_{Pup}}$ is coded alongside the dictionary $\mathbf{D_{Pup}}$ using a Huffman coding scheme (Huffman, 1952). The coding stage is a lossless compression strategy that seeks to represent the data with the lowest number of bits per symbol. More details on applying entropy coding techniques in the DL domain can be found in Faouzi Zizi and Turquais (2022).

Data reconstruction

First, we apply the inverse operations of the quantization and coding stages as described in detail by Faouzi Zizi and Turquais (2022) to obtain the reconstructed $\hat{\mathbf{D}}_{\mathbf{Pup}}$and $\hat{\mathbf{X}}_{\mathbf{Pup}}$. Then, we can easily reconstruct the total vertical particle velocity, the total pressure, the up-going pressure or the down-going pressure seismic shot gathers. For example: multiplying $\hat{\mathbf{D}}_{\mathbf{Pup}}$ and $\hat{\mathbf{X}}_{\mathbf{Pup}}$ will reconstruct the up-going pressure seismic shot gathers; multiplying $\hat{\mathbf{D}}_{\mathbf{Pup}}$ and $\hat{\mathbf{X}}_{\mathbf{Pup}}$, after rescaling the first K coefficients of the sparse vectors in $\hat{\mathbf{X}}_{\mathbf{Pup}}$ denoted $\hat{\mathbf{X}}_{\mathbf{Pup}}[1:K,:]$ by -1 following (16), will reconstruct the down-going pressure seismic shot gathers; multiplying the first K atoms of $\hat{\mathbf{D}}_{\mathbf{Pup}}$ denoted $\hat{\mathbf{D}}_{\mathbf{Pup}}[:, 1:K]$ and the first K coefficients of the sparse vectors in $\hat{\mathbf{X}}_{\mathbf{Pup}}$ denoted $\hat{\mathbf{X}}_{\mathbf{Pup}}[1:K,:]$, after rescaling $\hat{\mathbf{X}}_{\mathbf{Pup}}[1:K,:]$ by a factor of -2 following (16), would reconstruct the corrected velocity seismic shot gathers; finally, multiplying the last K atoms of $\hat{\mathbf{D}}_{\mathbf{Pup}}$ denoted $\hat{\mathbf{D}}_{\mathbf{Pup}}[:, K+1:2K]$ and the last K coefficients of the sparse vectors in $\hat{\mathbf{X}}_{\mathbf{Pup}}$ denoted $\hat{\mathbf{X}}_{\mathbf{Pup}}[K+1:2K,:]$, after rescaling $\hat{\mathbf{X}}_{\mathbf{Pup}}[K+1:2K,:]$ by a factor of 2, would reconstruct the pressure seismic shot gathers. Figure 4 shows the reconstructed up-going pressure sparse representation number 3 denoted $\hat{\mathbf{x}}_3^{\mathbf{Pup}}$, which corresponds to the combination of the velocity and pressure sparse representations $\hat{\mathbf{x}}_3^{V_z}$ and $\hat{\mathbf{x}}_3^{P}$, respectively, and the reconstructed dictionary $\hat{\mathbf{D}}_{\mathbf{Pup}}$ corresponding to the combination of the corrected velocity dictionary $\hat{\mathbf{D}}_{CV_z}$ and the pressure dictionary $\hat{\mathbf{D}}_P$, respectively. Multiplying $\hat{\mathbf{D}}_{\mathbf{Pup}}$ and $\hat{\mathbf{x}}_3^{\mathbf{Pup}}$gives the linear combination of atoms needed to reconstruct the up-going pressure patch number 3. Note that, as only the 256 first atoms of the 6000 atoms of each dictionary are displayed in Figure 4, more atoms are in fact needed to obtain the final up-going pressure field patch displayed in the figure.

DATA APPLICATION

Synthetic data

In this section, we will assess the capability of the PDL method for wavefield separation (WSPDL) method to correct for velocity obliquity scaling and to reconstruct the up- and down-going pressure fields. We have modelled a synthetic data set, where we have access to the true result, and compared our method to an optimized industry-standard FK method (FK-WS).

The total pressure and total vertical particle velocity wavefields were modelled using acoustic finite difference modelling for the 2D P-wave velocity model shown in Figure 5, which comprises a 200 m thick water layer and five sedimentary layers. The different layers have different boundary dips and velocities to create dipping events with different slopes. The density increases slightly with depth and goes from 1.96 g/cm^3 in the first sedimentary layer to 2.11 g/cm^3 in the fifth and last one (Figure 5). In addition, diffracting points have been added to create conflicting dips in the shot

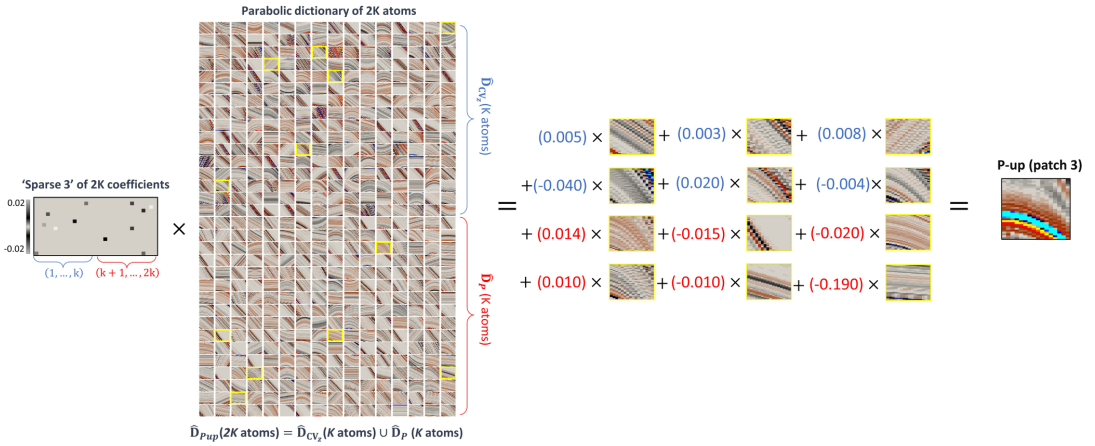

FIGURE 4 Reconstruction of the up-going pressure patch 3 by multiplying the up-going pressure dictionary, which is a combination of the corrected velocity and pressure dictionaries, and the up-going pressure sparse representation, which is a combination of the velocity and pressure sparse representations.

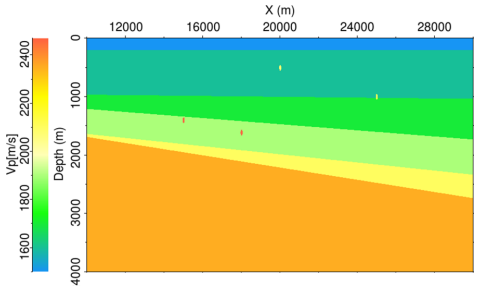

FIGURE 5 2D P-wave velocity model used to generate the synthetic data example of 100 shots. Four diffracting points are represented in the model at different depths with small oval shapes. The depth of the source is 6 m, and the depth of the cable is 20 m, where the receiver spread is fixed.

gathers which are challenging to reconstruct. The up-going parts of the pressure and vertical particle velocity fields were first modelled without free-surface effects. Then, the down-going parts of the pressure and vertical particle velocity fields, which are reflected at the sea surface, were modelled using virtual receivers located in a mirror-image position relative to the sea surface. Further, the direct arrival was also modelled and subtracted from the four data sets. We then combined the up- and down-going parts of the pressure and the up- and down-going parts of the vertical particle velocity wavefields to simulate the total pressure and the total vertical particle velocity measurements recorded by the hydrophones and the geophones, respectively. Hence, we have produced three data sets each of 100 shots: the input vertical particle velocity and pressure measurements, and their corresponding up-going pressure field which will be used as reference (true result).

The modelled data sets were sampled at 2 ms in time and 3.125 m in space. The depth of the source was 6 m, and the depth of the receivers was 20 m. Source ghosts and free-surface multiples were not modelled for the three data sets of 100 shots. Moreover, the modelled data sets underwent pre-processing, as shown in Figure 6, in preparation for the application of the wavefield separation process using the WSPDL and FK-WS methods. The performance of these methods is then evaluated and compared. Here, we consider three data sets: the recorded total vertical particle velocity, the recorded total pressure and the reference up-going pressure wavefield. First, we applied a 20 Hz low-cut filter to the three data sets because low frequencies are handled by other methods when dual-sensor wavefield separation is applied to field data examples. For low-frequencies, typically below 20 Hz, there is no decomposition of the wavefield into up- and down-going parts. Indeed, the velocity data are heavily contaminated by noise at the lowest frequencies. Thus, this problem is usually solved by the so-called low-frequency conditioning, where the noisy low frequency velocity data are reconstructed from the relatively clean pressure data (Day et al., 2013). Then, we applied a dip filter to remove steeply dipping events with apparent propagation velocities of less than 1550 m/s. This was done to allow a fair comparison of the WSPDL method and the industry-standard FK-WS method in terms of wavefield separation performance. Indeed, a tapering is generally applied at such dips in conventional FK-WS methods to avoid instability arising from the application of large, rapidly varying obliquity scaling factors. Finally, we decimate the data by a factor of 4 in the space dimension to obtain data

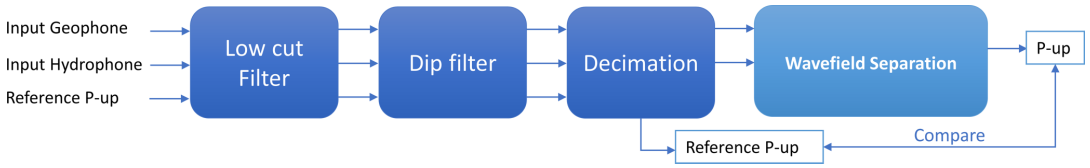

FIGURE 6 Pre-processing workflow for synthetic data before applying the dual-sensor wavefield separation process.

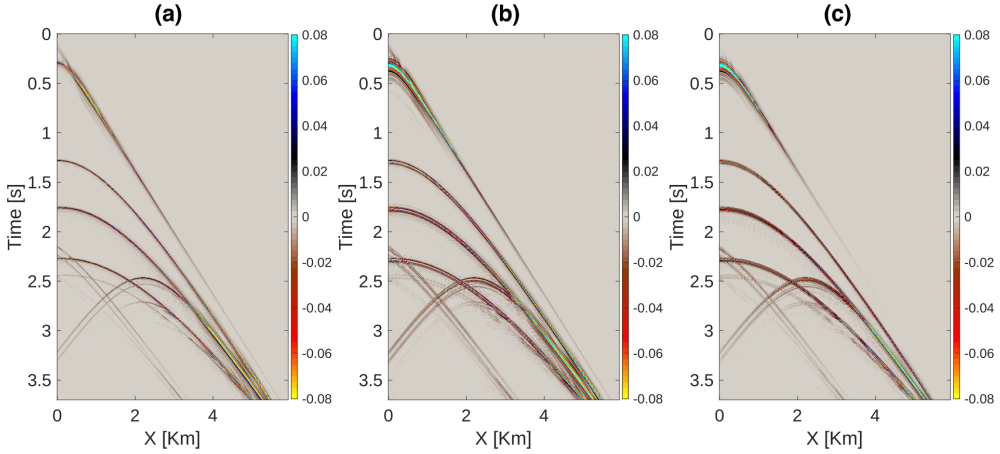

FIGURE 7 One seismic shot gather of the 100 shot synthetic data example: (a) reference up-going pressure field; (b) total pressure measured by the hydrophones and (c) total vertical particle velocity component measured by the geophones.

sets sampled at 12.5 m, normally employed for field data. Figure 7 shows the same example shot gather for the three data sets: the reference up-going pressure field (Figure 7a), the input pressure (Figure 7b) and the input velocity (Figure 7c) after the decimation step but before applying the dual-sensor wavefield separation process (Figure 6).

After pre-processing the data sets, we apply the wavefield separation process to the synthetic input pressure and velocity recordings and compare the resulting up-going wavefield to the reference up-going wavefield. The wavefield separation process was carried out using the WSPDL method, then using the optimized industry-standard FK-WS method without interpolation and a third time using the same FK-WS method after interpolating the data to a 3.125 m spatial sampling interval using an industry-standard interpolation algorithm to handle the aliased events (Bekara & Robin, 2015). In real field data cases, the FK-WS method is used after interpolation as the data are more commonly acquired at 12.5 m. Figure 8 shows the residual difference between the reference up-going pressure wavefield (Figure 8a) and the combination of the input pressure (Figure 7b), and velocity (Figure 7c) when: No obliquity scaling is applied (Figure 8b), the industry-standard FK-WS method is used without interpolation (Figure 8c), the industry-standard FK-WS method is used with interpolation (Figure 8d) and the WSPDL method as described in the methodology section is used (Figure 8e). Figure 9 shows the corresponding f–k spectra of the shot gathers displayed in Figure 8. We compute the signal-to-residual ratio (SRR) metric for the different methods following Equation (5), to assess the quality of the wavefield separation results, with $\mathbf{d}_{\text{orig}}$ being the reference up-going pressure field (Figure 8a) and $\mathbf{d}_{\text{rec}}$ the reconstructed up-going pressure field after applying one of the wavefield separation methods. Figures 8b and 9b show visually the results obtained in case an obliquity correction factor F of 1 is employed with Equation (8). In this case, we obtain a SRR value of 10.1 dB. If no interpolation is used prior to the industry-standard FK-WS method (Figures 8c and 9c), the aliased events are not well reconstructed and a SRR of 16.54 is reached. However, when the same FK-WS method is used after interpolation (Figures 8d and 9d) the residuals are negligible and a high level of SRR = 27.64 dB is reached. From Figure 8e, we can clearly see that the WSPDL method has also succeeded in reconstructing the up-going pressure

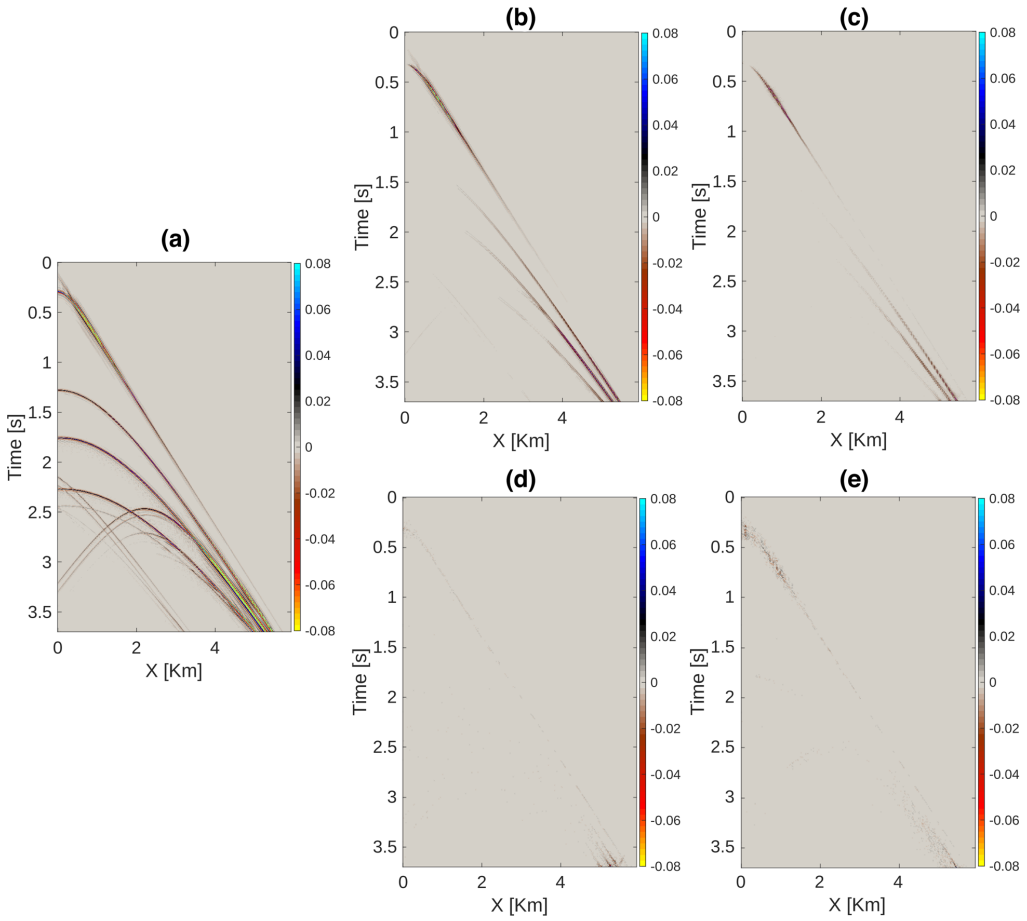

FIGURE 8 Wavefield separation results for the shot gather represented in Figure 7: (a) reference up-going pressure field; (b) residuals when no obliquity scaling is applied; (c) residuals after applying the FK-WS method without interpolation; (d) residuals after applying the FK-WS method with interpolation and (e) residuals after applying our WSPDL method.

field well and reaches a high level of SRR = 24.43 dB, and with only weak residuals associated with very high amplitude events. Thus, although the industry-standard FK-WS method (Figures 8d and 9d) achieves slightly better results when interpolation is included compared to the WSPDL method (Figures 8e and 9e), both methods achieve high quality wavefield separations. Moreover, our method has shown its effectiveness in reconstructing aliased events without the need for interpolation. That is because parabolic dictionary learning (PDL) reconstructs aliased events under its own set of assumptions, namely sparsity and parabolic constraint. Under these assumptions, PDL has shown the ability to provide better interpolation results compared to a standard FK-based interpolation method (Turquais et al., 2019). In addition, our method has accomplished the dual-sensor wavefield separation process in the dictionary learning (DL) compressed domain with CR = 13.13, that is the data used to reconstruct the up-going pressure field are 13 times smaller than the input velocity and pressure data sets. The reconstruction results obtained using the different methods are summarized in Table 1. In this synthetic data example, we have in fact compressed the input velocity and pressure data sets and then reconstructed them prior to using both FK-WS methods (with and without interpolation). This procedure was chosen to ensure that the data input to the FK-WS methods include any errors introduced by the compression step, thereby permitting a fair assessment of the effectiveness of the WSPDL method to correct for the obliquity scaling problem itself. The WSPDL method compresses the data and corrects for the obliquity scaling problem simultaneously in

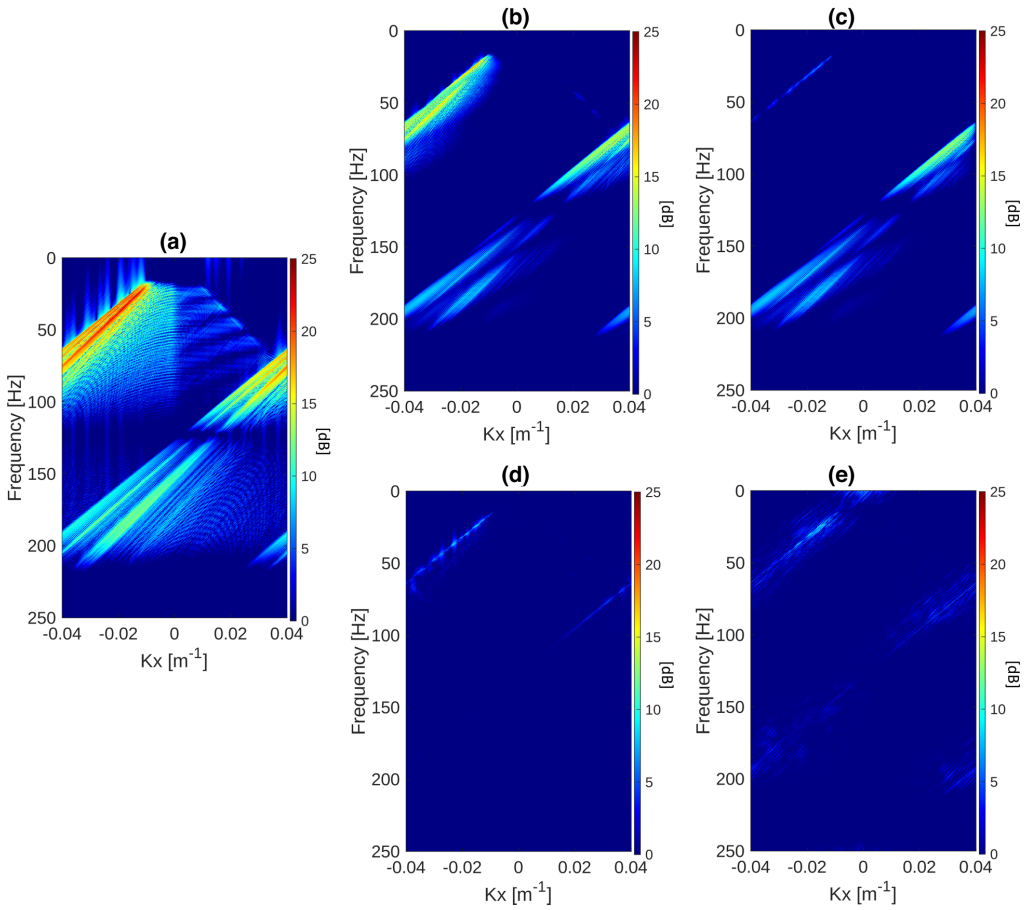

FIGURE 9 The *f–k* spectra of the wavefield separation results represented in Figure 8: (a) reference up-going pressure field; (b) residuals if no obliquity correction is applied; (c) residuals after applying the FK-WS method without interpolation; (d) residuals after applying the FK-WS method with interpolation and (e) residuals after applying our WSPDL method.

the compressed domain. Hence, it is not possible to apply the wavefield separation process without compressing the data when using WSPDL. If no compression and reconstruction are applied to the input velocity and pressure data sets, the results of the optimized industry-standard FK-WS method when used with interpolation reach a SRR level of 34.02 dB, and when no interpolation is used it reaches a level of SRR = 16.79 dB. The computational costs of the FK-WS method with interpolation and the WSPDL are comparable. However, when using WSPDL, the compression is the most expensive part, whereas the reconstruction is very efficient and fast as it requires only matrix multiplications as described earlier in the methodology section. This means that once the data are in the compressed domain, one can also reconstruct the down-going pressure field, the pressure or the corrected vertical particle velocity with almost no extra cost.

Field data

We now consider a more realistic field data set that consists of 3320 shots comprising a full 2D sail-line of marine seismic acquisition. The data were acquired in the 'Nordland ridge' area offshore Norway with a time sampling of 2 ms and a spatial sampling of 12.5 m. The number of time samples (Nt) and the number of channels (Nx) are $Nt = 2050$ and $Nx = 647$, respectively. For every shot, both the total pressure and the total vertical particle velocity were recorded. Both data recordings were pre-processed

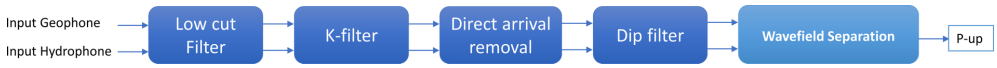

FIGURE 10 Different pre-processing steps before applying the dual-sensor wavefield separation to the field data example of 3320 shots.

TABLE 1 Comparison of the different wavefield separation methods in terms of signal-to-residual ratio (SRR), ability to reconstruct aliased data parts (+) or not (−), and compression ratio based on the synthetic data example

Method	WSPDL	Interpolation + FK	FK
SRR (dB)	24.43	27.64	16.54
Aliasing	+	+	−
CR	13.13	0.25	1

Note: We remind that compression/decompression is applied to the input data sets of all three methods to allow a fair comparison of the wavefield separation performance.
Abbreviations: CR, compression rate, SRR, signal-to-residual ratio.

before applying the dual-sensor wavefield separation process (Figure 10). Later, we will describe in detail the different steps of the pre-processing sequence. The purpose of the test is to assess the effectiveness of WSPDL when applied to field data. This is done by comparing its performance to the optimized industry-standard FK-WS method with interpolation, as this approach has been shown to provide high quality results in the synthetic data example. However, for the field data we do not have access to the true result. Moreover, we do not apply any compression to the input velocity and pressure field data sets prior to using the FK-WS method. This means that the combined effects of compression and wavefield separation are included in the quantitative assessment of the differences between the results of both methods. Furthermore, the shot gathers will be pre-stack migrated to assess the differences between both methods in the image domain.

Figure 10 shows the pre-processing steps applied to the input velocity and pressure data sets. First, a low-cut filter is applied to remove frequencies below 20 Hz for the same reason previously mentioned in the synthetic data example. Then, a *k*-filter or spatial matching filter is applied to match the responses of the geophones and hydrophones groups. This process is required because the distributions of individual sensors in the hydrophones and geophones groups are not identical. We then remove the direct arrival and apply a dip filter to remove steep events with apparent velocities less than 1550 m/s and limit our comparison to the signal cone as in the synthetic data example. Finally, the wavefield separation process is applied, once using the FK-WS method after interpolating the data to 6.25 m spatial sampling as generally done in production, and the second time using the WSPDL method. The two up-going pressure field data sets obtained are then compared. Here, we apply the same parameters described in the methodology section when using the WSPDL method. In fact, we developed our algorithm such that each group of 200 shots are processed independently. Hence, for each group of 200 shots, 2 dictionaries of 6000 atoms will be learned, one for the velocity and one for the pressure data sets, then both will be combined into one dictionary of 12,000 atoms. However, when the total number of shots cannot be divided by 200, the algorithm will automatically find the closest number to 200 that will make this division possible. For example in our case we deal with 3320 shots; therefore, groups of 208 shots will be processed independently, whereas the last set will comprise exactly 200 shots. The choice of 200 shots was found to be a good compromise between the accuracy of reconstruction and the sparsity level. Indeed, applying the same parameters previously used on a 100 shots data set to a 200 shot data set will lead to a higher level of sparsity and thus a higher compression rate (CR) as 12,000 atoms will now be used to reconstruct 200 shot gathers and not only 100. However, a very high number of shot gathers processed at the same time might lead to insufficient number of atoms to reconstruct the data with the high level of reconstruction SRR = 30 dB imposed earlier in the methodology section. Our aim is to develop a method which is transferable to different data sets without changing the DL parameters.

Figure 11 shows one shot gather of the resulting up-going pressure field after applying the wavefield separation process using both methods: the industry-standard FK-WS method (Figure 11a) and the WSPDL method (Figure 11b), respectively. Figure 11c shows the difference between the two shot gathers. The *f*–*k* spectra of Figure 11a,c are displayed in Figure 11d,f, respectively. The small differences between both methods are primarily due to locally strong amplitudes at near offsets. Indeed, this is a very challenging part of the data set for both methods where a rapid change of amplitudes occurs in both space and time. Here, the FK-WS method suffers from the truncation effect at zero offset, whereas the WSPDL method is based on the ray theory which assumes smoothly changing amplitudes along the travel time move outs (Bortfeld, 1989; Ursin, 1982). Moreover, in Figure 11f, we see that such differences are primarily concentrated at high frequencies (around 160 Hz), whereas the differences at the low frequencies (lower than 60 Hz) and low wavenumbers are relatively weak compared to the high energy of the primary even if it is significant relatively to the rest of the energy displayed in the *f*–*k* spectrum of the differences (Figure 11d–f). From Figure 11d,e, we note

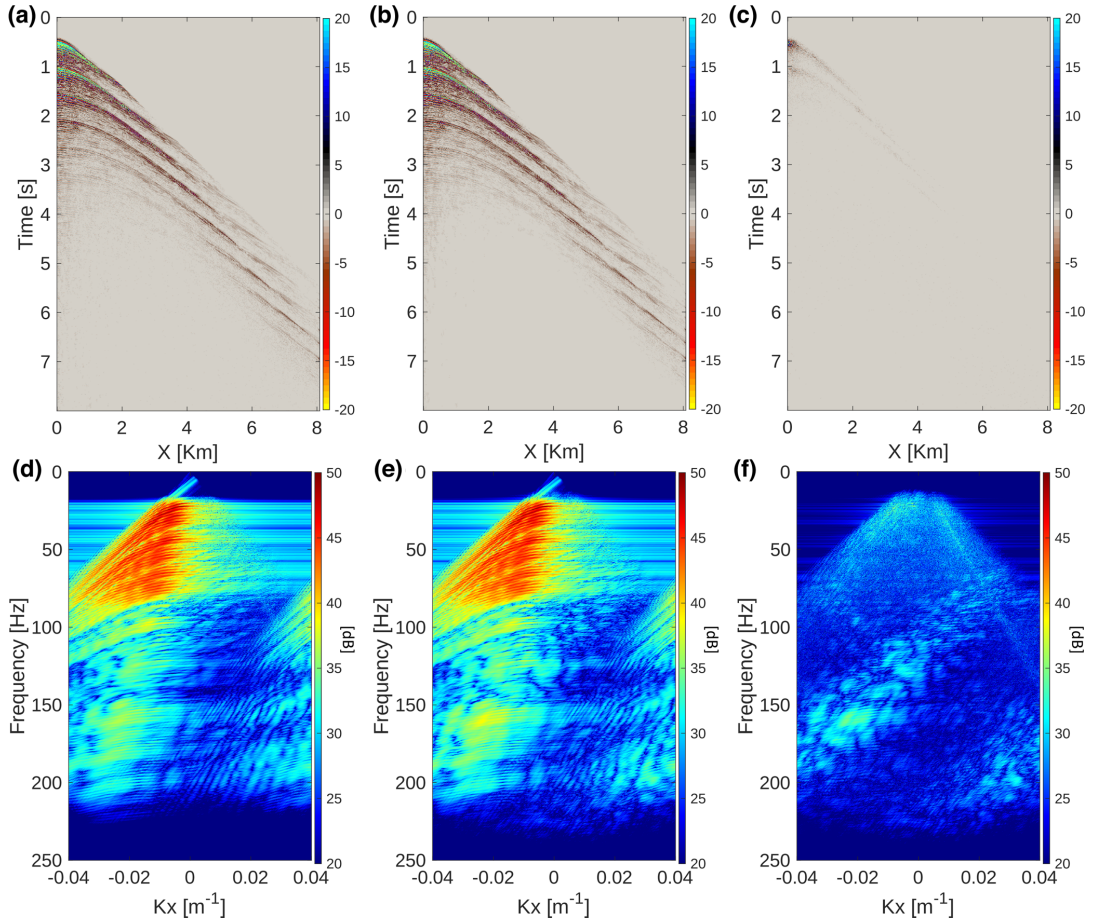

FIGURE 11 Dual-sensor wavefield separation results for one shot gather of the field data example: (a) up-going pressure field after applying the FK-WS method with interpolation; (b) up-going pressure field after applying the WSPDL method; (c) differences between (a) and (b) and (d–f) the *f*–*k* spectra of (a–c), respectively.

that the WSPDL method reconstructs the high frequency events around 160 Hz with more energy than the FK-WS method. Hence, such differences are most likely not due to the compression effects, but higher values of the obliquity correction factor F applied to the velocity data for such dipping events in case of WSPDL. We recall that we spatially interpolate the data with a factor of 2 when the industry-standard FK-WS method is applied, and that we have already demonstrated the ability of the WSPDL method to work beyond aliasing in the synthetic data example. Therefore, it is reasonable to observe higher amplitudes on highly dipping events when using the WSPDL at frequencies higher than 120 Hz which is the frequency limit beyond which spatial aliasing will adversely affect the output of the FK-WS method for spatial sampling of 6.25 m. However, the FK-WS method does not show specific problems above 120 Hz on the synthetic data example, because the synthetic data are sparse in frequency, which makes it easy for frequency-based interpolation methods such as the industry-standard interpolation algorithm (Bekara & Robin, 2015) used here, to reconstruct clean aliased events (Figure 9d) compared to the real data case. Note that the WSPDL method has reconstructed the up-going pressure field in the compressed domain with CR = 15.81.

After comparing the results of both wavefield separation methods on seismic shot gathers, we will now pre-stack migrate the resulting up-going pressure field data set of 3320 shots comprising a full sail-line of marine seismic acquisition. We applied a simple seismic processing workflow with main steps as summarized in Figure 12. This workflow does

FIGURE 12 Main steps of the seismic processing workflow to generate a 2D line image after applying the dual-sensor wavefield separation process using the FK and the WSPDL methods.

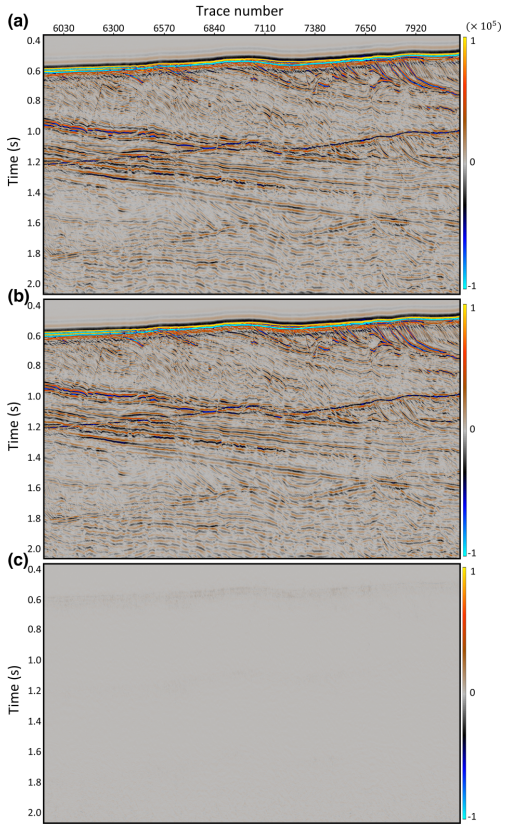

FIGURE 13 (a) 2D seismic image after applying the FK-WS method; (b) 2D seismic image after applying the WSPDL method and (c) difference between (a) and (b).

not replicate a full commercial processing but is sufficient to assess the impact of both wavefield separation methods on a 2D seismic image. Figure 13 shows the part of the 2D seismic image, where most dipping events occur. Here, the focus is on the shallow part as the deep part is heavily contaminated by noise. Visually, it is hard to discern any differences between Figure 13a,b, representing the 2D seismic images after applying the industry-standard FK and the WSPDL methods, respectively. The differences between both reconstructions are displayed in Figure 13c and are mainly located close to the water bottom reflector, where amplitudes are very strong. Hence, we can conclude that the two methods provide very similar results. However, in case of WSPDL, a compression factor of almost 16 is obtained and aliased energy is handled without the need for explicit interpolation. In order to confirm the visual results, we perform an amplitude spectrum analysis of the 2D seismic images, and their differences. Figure 14 shows the cumulative amplitude spectra of the images in Figure 13a–c, denoted in blue, green and red, respectively. This figure shows that the spectrum of the differences between both images, which is 10 dB to 30 dB below signal (in red), is relatively flat and has a dissimilar shape to that of the 2D seismic sections. The spectra of the images formed from the two up-going pressure fields (in green and blue) are very similar especially below 125 Hz. This analysis confirms that the FK and WSPDL methods provide very similar results. Moreover, we do not observe any significant loss of energy due to the compression effect when we use the WSPDL method knowing that the resulting image was reconstructed from data almost 16 times smaller in size than the original data used for the FK-WS method. That is because we have targeted a high level of reconstruction of SRR = 30 dB for the input velocity and pressure data sets. We further observe that the WSPDL method recovers events between 120 and 200 Hz with higher energy than the FK-WS method. However, this does not mean that one method is more correct than the other. Indeed, the water bottom differences are located at these frequencies which is in accordance with the analysis of the shot gathers results shown in Figure 11.

DISCUSSION

Compression sensitivity analysis

Although we have already discussed our choices for the different dictionary learning (DL) parameters, we should still discuss the compression impact on the wavefield separation results when using the PDL method for wavefield separation (WSPDL) method. Indeed, a high input signal-to-residual ratio (SRR) level of 30 dB has been imposed as a target to allow an accurate reconstruction of the input velocity and pressure data sets after compression. Such an input SRR level (that we will denote SRR_{in}) has allowed a good reconstruction of the up-going pressure field for both the synthetic data example where the SRR after wavefield separation (that we will denote SRR_{WS}) reached a high level of 24.43 dB with

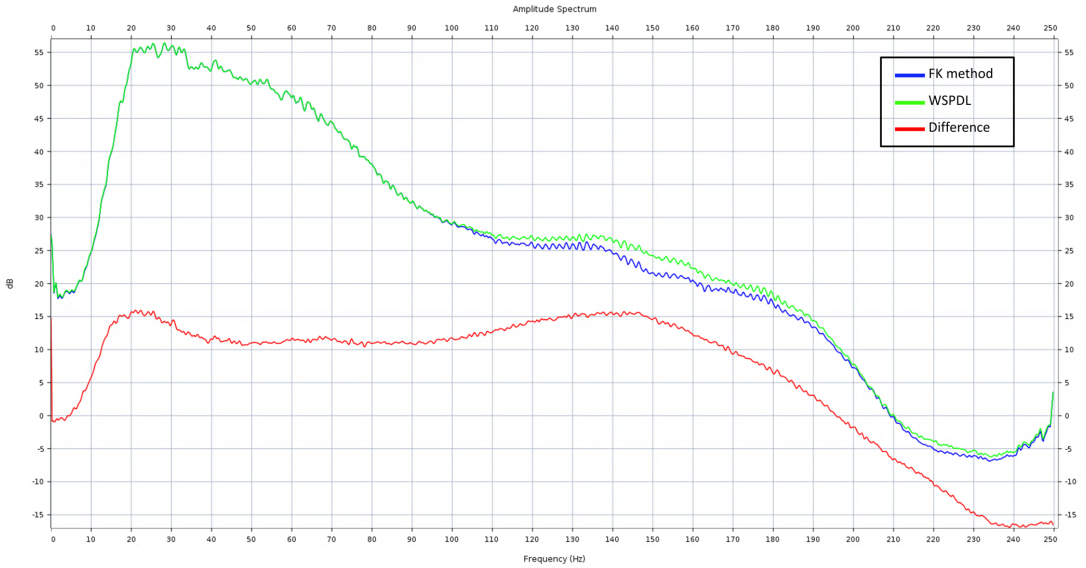

FIGURE 14 Amplitude spectrum analysis of the 2D seismic images represented in Figure 13.

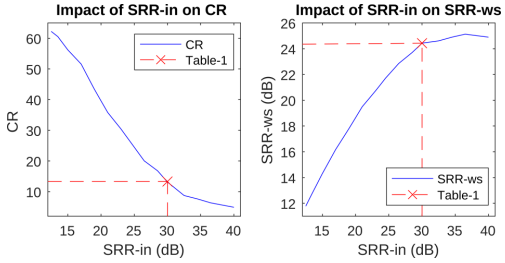

FIGURE 15 Impact of different input signal-to-residual levels (SRR_{in}) on the compression ratio (CR) and the wavefield separation reconstruction levels (SRR_{WS}). The red cross on the graphs corresponds to the case reported in Table 1, where SRR_{in} = 30 dB.

compression rate (CR) = 13.13 and the field data example where a 2D line section was reconstructed with insignificant energy loss while reaching a CR of almost 16. Acknowledging that as we increase the SRR_{in}, we decrease the CR, we introduce a simple experiment in which the WSPDL method is tested with different values of SRR_{in}. We then observe how the CR level impacts the wavefield separation results via the SRR_{WS} employing the 100 shot synthetic data set. The graphs displayed in Figure 15 show that the WSPDL method is stable as the CR level is decreasing when the SRR_{in} is increasing as expected. We also observe from the same figure that the SRR_{WS} level becomes relatively stable (between 24 and 25 dB) when SRR_{in} is higher than 30 dB, which shows the reconstruction limits of our method. Indeed, targeting a higher level of SRR_{in} = 36.5 dB in our method could lead to a slightly better SRR_{WS} level (Figure 15), but the CR level would drop from 13.13 to 6.35. In Figure 16 we display four different up-going pressure field reconstructions given different SRR_{in} levels. Here, we observe good performance in terms of up-going pressure field reconstructions when CR = 6.35 and CR = 13.13 (Figure 16a,b,e,f) and one can hardly see any differences between those reconstructions. When CR is around 26 (Figure 16c,g), few low amplitude events are not well reconstructed, and residuals are uniformly distributed along the different events below the first reflection. However, even when reaching such a high CR level the visual quality of the reconstruction is still acceptable. In Figure 16d,h, a high level of CR = 43.36 is reached, and many events are not well reconstructed. These displays validate our choice of targeting a high SRR_{in} level of 30 dB, as it allows to reach both high SRR_{WS} and CR levels.

Future work

In the synthetic data and field data subsections, we have applied the same pre-processing steps (Figures 6 and 10) to input data sets before applying the WSPDL or the optimized industry-standard FK-WS method. Indeed, this paper focuses only on enabling the dual-sensor wavefield separation process in the DL compressed domain. Thus, it is important to apply the same pre-processing steps before applying both

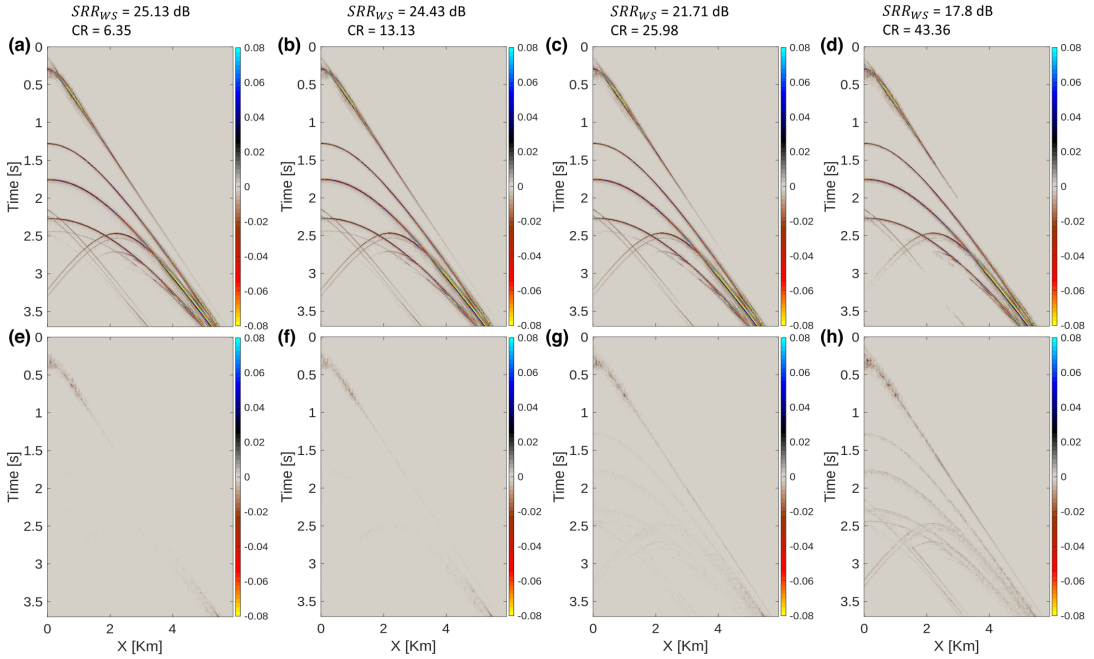

FIGURE 16 Compression effect on the 2D synthetic up-going pressure field reconstruction results: (a–d) up-going pressure field reconstructions after applying the WSPDL method with different compression levels and (e–h) the residuals corresponding to (a–d) after subtracting the reference up-going pressure field.

methods to allow a fair comparison in terms of wavefield separation performance. However, some of these steps are more suited for FK-WS methods as they already include a transformation in the FK domain. Hence, future work can focus on enabling pre-processing or other processing steps in the DL compressed domain to avoid going back and forth between different processing domains. For example the dip filter does not necessarily need to be applied before the WSPDL method here. This was applied in synthetic and field data examples to fairly compare the impact of both methods on the signal cone as tapering is generally applied in conventional FK-WS methods as mentioned in the data application section. In practice, a taper can also be applied after reconstructing the data with the WSPDL method or a dip filter can directly be applied at the atoms level using the kinematic parameters. Another example is the *k*-filter that was not needed to compare both methods but was applied in the real data example to simulate a real data case where the responses of the geophones and hydrophones groups need to be matched. Consequently, enabling such a filter in the DL compressed domain can be very convenient. Moreover, the low-cut filter was applied because low frequency velocity data are generally reconstructed from the relatively clean pressure data (Day et al., 2013), as mentioned earlier in the data application section. Therefore, one can also investigate the possibility of using the low frequencies pressure data to reconstruct the low frequencies up-going pressure field directly in the DL compressed domain.

In this paper, we have learned the dictionaries $\mathbf{D}_{V_z}$ and $\mathbf{D}_P$ independently. This was enough to achieve good reconstruction of the up-going pressure field while reaching a high level of compression. Instead, one could use a joint DL method (Wang et al., 2022) where the same kinematic parameters are imposed to the atoms of both dictionaries. Such an approach may lead to better approximations of the obliquity correction factors, as the total pressure data will also be used in the estimation of the kinematic parameters. Furthermore, the scope of this paper focuses on applying WSPDL in the 2D case. The extension to the 3D data case can be done by learning 3D atoms which would enable the reconstruction of the 3D wavefield. 3D atoms would be characterized by five parameters instead of only two in the 2D case, namely two slopes, two curvatures and a mixed travel time parameter correlating both directions (in-line and crossline). In the common reflection surface approach, Hoecht et al. (2009) estimated these parameters to achieve 3D interpolation. Similarly, such parameters can be learned using WSPDL. Alternatively, one can adopt a pseudo-3D approach where 2D dictionaries

will be learned independently in the in-line and crossline directions. Such an approach would be computationally cheaper than learning 3D atoms but might lead to less accurate results as correlations between the in-line and crossline directions will not be used in the estimation of the kinematic parameters. In this case, one should still think about methods to correlate the in-line and crossline dictionaries such as joint DL methods (Wang et al., 2022). Moreover, the paper focuses also on the dual-sensor towed streamer acquisition (or two-components streamer data acquisition), where: One hydrophone and one geophone are used to record the total pressure and the total particle velocity motion, respectively. Hence, other acquisition settings do exist such as the 3C streamer data acquisition, where two geophones or accelerometers record the particle velocity field in the vertical and the cross-cable directions. However, the horizontal component is heavily contaminated by noise in the in-line direction. The 3C acquisition setting would theoretically provide the crossline component of the particle velocity measurement without the need to learn kinematic parameters in the crossline direction (Vassallo et al., 2014). Such information can be combined with the kinematic parameters estimated in the in-line direction via WSPDL to allow a better approximation of the reconstructed wavefield.

CONCLUSION

In this work, we have successfully demonstrated a novel method that enables dual-sensor wavefield separation in a compressed domain using a parabolic dictionary learning (PDL) algorithm. The method (WSPDL) uses PDL to transform input data sets, namely the total pressure and the total vertical particle velocity measurements, to a compressed domain composed of two main parts: a dictionary of parabolic atoms and a set of sparse coefficients. The atoms of the dictionary are characterized by kinematic parameters such as the slope and the curvature, which are used to correct for velocity obliquity scaling directly in the compressed domain, and thus allow to successfully separate the up- and down-going parts of the pressure wavefield. We have tested and validated the performance of our method by quantifying the accuracy of the up-going pressure field reconstruction using the signal-to-residual ratio metric based on a 100 shot synthetic data example, where we have access to the true result. Moreover, we observe similar wavefield separation performance when compared to an optimized industry-standard FK-WS method based on both synthetic and field data examples. Hence, we have observed only small differences between the two methods after 2D line pre-stack migration. Moreover, the WSPDL method is robust with respect to spatial aliasing without the need for data preconditioning such as interpolation and comes with the advantage of a data compression rate higher than 15. Such a method could allow full bandwidth data transfer from vessels to onshore processing centres as it reaches significant compression levels. The transmitted compressed data can be used to reconstruct not only the input data sets recorded by the hydrophones and the geophones, but also the up- and down-going parts of the wavefield without the need to run the conventional dual-sensor wavefield separation process as this reconstruction requires only a simple matrix multiplication.

In this paper, we have succeeded in applying one seismic processing step, namely the dual-senor wavefield separation, in the dictionary learning compressed domain using kinematic parameters. Enabling other seismic processing steps in the compressed domain using these kinematic parameters would provide a novel and efficient workflow to simultaneously compress and partly or fully process seismic data sets.

ACKNOWLEDGEMENTS

We would like to express our gratitude to the R&D team of PGS for the fruitful discussions on the subject and thank PGS for providing the data and giving the permission to publish this work.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are confidential, not publicly available and cannot be released due to privacy or ethical restrictions.

REFERENCES

Aharon, M., Elad, M. & Bruckstein, A. (2006) K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54, 4311–4322.

Amundsen, L. (1993) Wavenumber-based filtering of marine point-source data. *Geophysics*, 58, 1335–1348.

Averbuch, A., Coifman, R., Donoho, D., Israeli, M. & Walden, J. (2001) Fast slant stack: a notion of radon transform for data in a cartesian grid which is rapidly computible, algebraically exact, geometrically faithful and invertible. *SIAM Scientific Computing*, 37, 192–206.

Beckouche, S. & Ma, J. (2014) Simultaneous dictionary learning and denoising for seismic data. *Geophysics*, 79, A27–A31.

Bekara, M. & Robin, F. (2015) Seismic data up-sampling beyond aliasing using polynomial phase signals. In: *77th EAGE Conference and Exhibition*. pp. 1–5.

Bortfeld, R. (1989) Geometrical ray theory: rays and traveltimes in seismic systems (second-order approximations of the traveltimes). *Geophysics*, 54, 342–349.

Bruckstein, A.M., Donoho, D.L. & Elad, M. (2009) From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51, 34–81.

Day, A., Klüver, T., Söllner, W., Tabti, H. & Carlson, D. (2013) Wavefield-separation methods for dual-sensor towed-streamer data. *Geophysics*, 78, WA55–WA70.

Donoho, D.L. & Elad, M. (2003) Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell 1$ minimization. *Proceedings*

of the National Academy of Sciences of the United States of America, 100, 2197–2203.

Donoho, D.L. (2006) Compressed sensing. *IEEE Transactions on Information Theory*, 52, 1289–1306.

Duval, L.C. & Rosten, T. (2000) Filter bank decomposition of seismic data with application to compression and denoising. In: 70th SEG Annual International Meeting, Expanded Abstracts. pp. 2055–2058.

Elad, M. (2010) *Sparse and redundant representations: from theory to applications in signal and image processing*, 1st edition, New York: Springer Publishing Company.

Fajardo, C., Reyes, O.M. & Ramirez, A. (2015) Seismic data compression using 2D lifting-wavelet algorithms. *Ingenieria y Ciencia*, 11, 221–238.

Faouzi Zizi, M.O. & Turquais, P. (2022) A dictionary learning method for seismic data compression. *Geophysics*, 87, V101–V116.

Helal, E.B., Saad, O.M., Hafez, A.G., Chen, Y. & Dousoky, G.M. (2021) Seismic data compression using deep learning. *IEEE Access: Practical Innovations, Open Solutions*, 9, 58161–58169.

Herrmann, F.J., Calvert, A.J., Hanlon, I., Javanmehri, M., Kumar, R., van Leeuwen, T. et al. (2013) Frugal full-waveform inversion: from theory to a practical algorithm. *The Leading Edge*, 32, 1082–1092.

Hoecht, G., Ricarte, P., Bergler, S. & Landa, E. (2009) Operator-oriented CRS interpolation. *Geophysical Prospecting*, 57, 957–979.

Hubral, P., Schleicher, J. & Tygel, M. (1992) Three-dimensional paraxial ray properties: Part I. Basic relations. *Journal of Seismic Exploration*, 1, 265–279.

Huffman, D.A. (1952) A method for construction of minimum redundancy codes. *Proceedings of the IRE*, 40, 1098–1101.

Lin, T.T. & Herrmann, F.J. (2013) Robust estimation of primaries by sparse inversion via one-norm minimization. *Geophysics*, 78, R133–R150.

Mairal, J., Bach, F., Ponce, J. & Sapiro, G. (2009) Online dictionary learning for sparse coding. In: *26th Annual International Conference on Machine learning*. pp. 689–696.

Mallat, S. (2008) *A wavelet tour of signal processing: the sparse way*, 3rd edition, Cambridge: Academic Press.

Mansour, H., Wason, H., Lin, T.T., T, T. & Herrmann, F.J. (2012) Randomized marine acquisition with compressive sampling matrices. *Geophysical Prospecting*, 60, 648–662.

Pati, Y.C., Rezaiifar, R. & Krishnaprasad, P.S. (1993) Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*. pp. 40–44.

Rubinstein, R., Zibulevsky, M. & Elad, M. (2008) *Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. Technion Report, CS-2008-08*.

Schiavon, A.P., Ribeiro, K., Navarro, J.P., Vieira, M.B. & Silva, P.M.C. (2020) 3D poststack seismic data compression with a deep autoencoder. *IEEE Geoscience and Remote Sensing Letters*, 19, 1–5.

Söllner, W., Day, A. & Tabti, H. (2008) *Space Frequency Domain Processing of Irregular Dual-Sensor Towed Streamer Data. SEG expanded abstracts*, vol. 27. pp. 1078–1083.

Turquais, P., Asgedom, E.G., Söllner, W. & Gelius, L.-J. (2018) Parabolic dictionary learning for seismic wavefield reconstruction across the streamers. *Geophysics*, 83, V263–V282.

Turquais, P., Pedersen, M. & Söllner, W. (2019) Parabolic dictionary learning: A method for seismic data reconstruction beyond the linearity assumption. In: *81st EAGE Conference and Exhibition*. pp. 1–5.

Ursin, B. (1982) Quadratic wavefront and traveltime approximations in inhomogeneous layered media with curved interfaces. *Geophysics*, 47, 1012–1021.

Vassallo, M., Özbek, A., Kamil, Y., van Manen, D., Eggenberger, K. (2014) Reconstruction of signals from highly aliased multichannel samples by generalized matching pursuit. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. pp. 2382–2385.

Wang, X., Teng, Y., Gao, M. & Jiang, H. (2004) Seismic data compression based on integer wavelet transform. *Acta Seismologica Sinica*, 17, 123–128.

Wang, Y., Zhang, G., Li, H., Yang, W. & Wang, W. (2022) The high-resolution seismic deconvolution method based on joint sparse representation using logging–seismic data. *Geophysical Prospecting*, 70, 1313–1326.

Zhang, Y., Bergler, S. & Hubral, P. (2001) Common-reflection-surface (CRS) stack for common offset. *Geophysical Prospecting*, 49, 709–718.

Zheng, F. & Liu, S. (2012) *A fast compression algorithm for seismic data from non-cable seismographs*. World Congress on Information and Communication Technologies, Trivandrum, India, 2012, pp. 1215–1219. doi: 10.1109/WICT.2012.6409260.

How to cite this article: Faouzi Zizi, M.O., Turquais, P., Day, A., Pedersen, M.W. & Gelius, L.J. (2023) Dual-sensor wavefield separation in a compressed domain using parabolic dictionary learning. *Geophysical Prospecting*, 1–19. https://doi.org/10.1111/1365-2478.13348

Paper III

Low frequency seismic deghosting in a compressed domain using parabolic dictionary learning

Mohammed Outhmane Faouzi Zizi [1,2], **Pierre Turquais**[1], **Anthony Day**[1], **Morten W. Pedersen**[1], **Leiv J. Gelius**[2]

Submitted for publication in *Geophysical prospecting*,

[1]PGS Geophysical AS, Oslo, Norway.
[2]University of Oslo, Department of Geosciences, Oslo, Norway.

III