

# Machine learning and artificial neural networks for seismic processing

Thomas de Jonge

Thesis for the degree of Philosophiae Doctor (PhD)  
University of Bergen, Norway  
2023

UNIVERSITY OF BERGEN



# Machine learning and artificial neural networks for seismic processing

Thomas de Jonge



Thesis for the degree of Philosophiae Doctor (PhD)  
at the University of Bergen

Date of defense: 05.05.2023



© Copyright Thomas de Jonge

The material in this publication is covered by the provisions of the Copyright Act.

Year: 2023

Title: Machine learning and artificial neural networks for seismic processing

Name: Thomas de Jonge

Print: Skipnes Kommunikasjon / University of Bergen



I dedicate this Ph.D. thesis to my fiance, Vilde,  
my son, Magnus,  
and our future children.

# Acknowledgements

Mainly, I would like to thank my supervisors, Dr. Vetle Vinje and Prof. Einar Iversen. This Ph.D. would not have been possible without your help. Thank you for your guidance through three years of hard work. Thank you, Vetle Vinje, for the long and exciting discussions that eventually led to new ideas and methods. I also appreciate your dedication and time invested in my research. You have been an invaluable resource during my Ph.D. Thank you, Einar Iversen, for all the help with academic research and writing. I would also like to thank Dr. Thomas Elboth, who, together with Vetle Vinje and Einar Iversen, recruited me for this Ph.D. position.

In addition, I would like to thank CGG Services Norway AS for providing me with software, hardware, seismic data, and a professional work environment. Mr. Gordon Poole, Dr. Peng Zhao, and Dr. Song Hou, thank you for helping me as co-authors to create results that eventually made it into my papers.

Thank you, friends and coworkers, at CGG and UiB for interesting discussions, lunch breaks, dinners, and social events. Especially thanks to my Ph.D. colleagues at CGG, Volodya Hlebnikov, Saskia Tschache, and Jing Sun. I would also like to thank Dr. Jan Erik Lie, Dr. Aina Juell Bugge, and the Ph.D. students from Lundin, for the bi-weekly meetings and discussions.

For financial support, I would like to thank the University of Bergen, CGG Services Norway AS, and the Norwegian Research Council (grant number 305450) for funding my Ph.D.

Finally, I would like to thank my family for your support. Mainly, I would like to thank my fiance, Vilde, for always believing in me. My son, Magnus, thank you for giving me extra motivation.

Thomas de Jonge  
Bergen, February 2023



# Preface

During the final year of my master's thesis, an oil company offered me a permanent position. I was happy with this job offer because the company and the work sounded quite good. However, I also had a small wish to pursue a Ph.D. position. I applied for an industrial Ph.D. position in CGG before I finished my master, wondering if I wanted the position. Luckily, CGG offered me the position. However, I was still uncertain if I should turn down the job in the Oil company for the Ph.D. position. Luckily, my family persuaded me to pursue the Ph.D. position. Looking back, I am glad I took the opportunity.

This thesis is submitted in fulfillment of the requirements for the degree of Philosophiae Doctor (Ph.D.) at the University of Bergen. The thesis is a collection of three papers. The research presented here was part of an Industrial PhD-scheme between the University of Bergen and CGG Services Norway AS, who, together with the Norwegian Research Council, funded the Ph.D. The granted Ph.D. project (305450) was supervised by Dr. Vetle Vinje, Senior Scientist at CGG Services Norway AS, and Prof. Einar Iversen from the University of Bergen. From September 2019 – December 2022, I was employed by CGG Services Norway AS. All the project work was conducted in the offices of CGG and at the University of Bergen.



# Abstract in English

The recent availability of powerful GPUs and open-source software have enabled artificial neural networks to solve several practical and industrial problems. We can apply neural networks to critical seismic processing steps such as swell noise attenuation, seismic interference attenuation, debubbling, deghosting, and deblending. Today, many of these processing steps involve significant testing and computational time. A neural network has the potential of reducing testing and computational time. In addition, neural networks can produce good results and be robust to changes in the input data. During the last decade, significant advancements in neural networks structures have been made and this development will most likely increase in the future.

An array of air guns is one of the most used sources in a marine seismic acquisition. Air guns have many advantages, which is why they are so commonly used. However, there are a few challenges when using air guns. The air-gun release an air bubble that oscillates and creates bubble noise after the first expansion. The bubble is quite strong compared to the peak, but combining multiple air guns can attenuate much of the bubble noise. However, a significant amount of bubble noise is still left in the far-field source signature, which elongates the signature and disturbs the seismic image. Modeling or estimating the bubble noise and removing it is possible but challenging because of the complex interaction between each air bubble. To add more complexity, any change in one air bubble will also change the other air bubbles and consequently change the bubble noise observed in the far-field signature. As a result, weather, relative positions of air guns, air gun depth, swell waves, air gun pressure, and air guns not firing can change the source signature. Therefore, the signature could change from shot to shot over a full survey. Some methods try to estimate the signature on each shot. However, this requires significant quality control and testing time.

The seismic waves reflected from the sea surface are called seismic ghosts. The sea surface reflection coefficient is close to -1, meaning that the ghost arrives with a polarity shift and a small time delay. In total, we observe three ghosts, the source ghost, the receiver ghost, and the source-receiver ghost. Ghosts are problematic because they elon-



gate and distort the seismic signal and cause notches in the frequency spectrum, reducing the temporal resolution. Therefore, it is important to remove the ghosts to improve the resolution and signal-to-noise ratio, which is important before geological interpretation or seismic inversion. We can use different acquisition geometries to attenuate the ghosts. However, the acquisition geometry alone cannot solve the deghosting problem. Therefore, processing methods have been developed to attenuate the ghost. One disadvantage of many conventional methods is that they require knowledge of the source, receiver positions, and sea surface reflection coefficient. However, marine seismic surveys include swell waves, weather, and incorrect source and receiver positions. Many conventional methods are, therefore, sensitive to any of these factors.

In this thesis, I describe an alternative approach using a convolutional neural network for debubbling and deghosting. For debubbling, I trained a network on real data containing an extensive range of source signatures to make the network robust to signature variations. If the signature in the prediction data is equal to one of the signatures in the training data, the network performs well. In addition, the network can adapt to a change in the signature in the middle of a sail line. Moreover, if the test data have similar geology to the training data, the network performs better than if not. I used real data from two locations on the Norwegian Continental Shelf to test this method. However, for deghosting, I created training data using demigration of stacked depth migrated images into shot gathers with and without ghosts. I need the source and receiver positions and the sea surface reflection coefficient to create demigrated data with ghosts. I perturb these parameters to generate variability in the ghost model, which makes the network more robust. On synthetic data, I demonstrate the robustness of the new method to variations in swells and sea-surface reflection coefficients. I have developed a method for pressure-only deghosting and dual-component deghosting. Both work well on real data when compared to conventional deterministic deghosting based on least-squares inversion in the  $\tau - p$  domain and P-Vz sum.

# Abstract in Norwegian

Nylig har tilgjengeligheten av kraftige GPUer og ”open source”-programvare gjort det mulig for kunstige nevralt nettverk å løse flere praktiske og industrielle problemer. Vi kan bruke nevralt nettverk til viktige seismiske prosesseringstrinn som dønning-støydemping, seismisk interferensdemping, debobling, deghosting og deblending. I dag krever mange av disse prosesseringstrinnene betydelig testing og beregningstid som nevralt nettverk har potensialet til å redusere. I tillegg kan nevralt nettverk gi gode resultater og takle endringer i innsamlingsdataene. I løpet av det siste tiåret har det blitt gjort betydelige fremskritt i å konstruere nevralt nettverk, og denne utviklingen vil mest sannsynlig øke i fremtiden.

Luftkanoner er en av de mest brukte kildene i en marin seismikkinnsamling. Luftkanoner har mange fordeler framfor andre kilder, som er grunnen til at de er så ofte brukt. Det er imidlertid noen utfordringer ved bruk av luftkanoner. Luftkanonen slipper ut en luftboble som svinger mellom å utvide seg eller å trekke seg sammen. Dette skaper boblestøy etter den første utvidelsen. Boblestøyen er ganske sterk, men ved å kombinere flere luftkanoner kan vi dempe mye av boblestøyen. Det er imidlertid fortsatt en betydelig mengde boblestøy igjen i fjernfeltkildesignaturen, noe som forlenger signaturen og forstyrrer det seismiske bildet. Modellering og estimering av boblestøyen og fjerning av støyen er mulig, men utfordrende på grunn av det komplekse samspeillet mellom hver luftboble. For å gjøre det mer komplekst, vil enhver endring i en luftboble også endre de andre luftboblene og følgelig endre boblestøyen observert i fjernfeltsignaturen. Som et resultat kan været, relative posisjoner til luftkanoner, luftkanondybde, dønninger, luftkanontrykk og luftkanoner som ikke skyter, endre kildesignaturen. Derfor kan signaturen endre seg fra skudd til skudd i løpet av en seismisk innsamling. Noen metoder prøver å estimere signaturen til et hvert skudd. Dette krever imidlertid betydelig kvalitetskontroll og testtid.

De seismiske bølgeene som reflekteres fra havoverflaten kalles seismiske ”ghosts”. Havoverflatens refleksjonskoeffisient er nær -1, noe som betyr at en ghost ankommer en hydrofon med et polaritetsskift og en liten tidsforsinkelse. Totalt observerer vi tre ghosts, kilde-

ghost, mottaker-ghost og kilde-mottaker-ghost. Ghosts er problematiske fordi de forlenger og forvrenger det seismiske signalet og forårsaker hakk i frekvensspekteret, noe som reduserer oppløsningen. Derfor er det viktig å fjerne ghosts for å forbedre oppløsningen og signal-til-støy-forholdet, noe som er viktig før geologisk tolkning eller seismisk inversjon. Vi kan bruke forskjellige innsamlingsgeometrier for å dempe ghosts. Likevel kan ikke innsamlingsgeometrien alene løse ghosts-problemet. Derfor er det utviklet prosesseringsmetoder for å dempe ghosts. En ulempe med mange konvensjonelle metoder er at de krever kunnskap om kildeposisjonen, mottakerposisjoner og havoverflatens refleksjonskoeffisient. I en marin seismisk undersøkelse vil vi oppleve dønningsbølger, vær og feil kilde- og mottakerposisjoner. Mange konvensjonelle metoder er følsomme for noen av disse faktorene.

I denne oppgaven beskriver jeg en alternativ tilnærming ved bruk av et konvolusjonelt nevralt nettverk for debobling og deghosting. For debobling trente jeg et nettverk på ekte data som inneholdt et omfattende utvalg av kilde-signaturer for å gjøre nettverket robust for signaturvariasjoner. Hvis signaturen i prediksjonsdataene er lik en av signaturene i treningsdataene, fungerer nettverket bra. I tillegg kan nettverket tilpasse seg en endring i signaturen midt i en seillinje. Dessuten, hvis testdataene har lignende geologi som treningsdataene, yter nettverket bedre. Jeg brukte ekte data fra to steder på norsk sokkel for å teste denne metoden. For deghosting laget jeg treningsdata ved å bruke demigrering av et dybdemigrert bilde til skuddsamlinger med og uten ghosts. Jeg trenger kilde- og mottakerposisjonene og havoverflatens refleksjonskoeffisient for å lage demigrerte data med ghosts. Jeg forandrer disse parameterne for å skape variasjon i ghosts-modellen, noe som gjør nettverket mer robust. På syntetiske data demonstrerer jeg robustheten til den nye metoden overfor variasjoner i dønninger og hav-overflate-refleksjonskoeffisienter. Jeg har utviklet en metode for kun trykk-deghosting og to-komponent deghosting. Begge fungerer godt på reelle data sammenlignet med konvensjonell deterministisk deghosting basert på minste kvadraters inversjon i  $\tau - p$  domenet og P-Vz summering.

# List of publications

1. T. de Jonge, V. Vinje, G. Poole, S. Hou, and E. Iversen, *Debubbling seismic data using a generalized neural network*, *Geophysics* **87**, 1, 2022.
2. T. de Jonge, V. Vinje, P. Zhao, G. Poole, and E. Iversen, *Source and receiver deghosting by demigration-based supervised learning*, *Geophysical Prospecting* **1**, 1, 2022 (published online).
3. T. de Jonge, V. Vinje, G. Poole, P. Zhao, and E. Iversen, *Deghosting dual-component streamer data using demigration-based supervised learning*, *Geophysical Prospecting* **1**, 1, 2022 (submitted to *Geophysical Prospecting*).

The first published paper is reprinted with permission from SEG. All rights reserved. Papers 2 and 3 are published under Open Access license CC-BY 4.0.



# Contents

Acknowledgements	i
Preface	iii
Abstract in English	v
Abstract in Norwegian	vii
List of publications	ix
<b>1 Introduction</b>	<b>1</b>
1.1 Outline . . . . .	1
1.2 Background . . . . .	2
1.3 Motivation . . . . .	4
1.4 Problem . . . . .	5
1.5 Scope . . . . .	5
<b>2 Debubbling seismic data</b>	<b>7</b>
2.1 Far-field source signature . . . . .	7
2.2 Debubbling . . . . .	10

---

<b>3</b>	<b>Deghosting seismic data</b>	<b>13</b>
3.1	Seismic ghosts . . . . .	13
3.2	Deghosting of pressure data . . . . .	15
3.3	Multi-component deghosting . . . . .	17
<b>4</b>	<b>Migration and demigration</b>	<b>21</b>
4.1	Migration . . . . .	21
4.2	Demigration . . . . .	23
<b>5</b>	<b>Machine learning and artificial neural networks</b>	<b>25</b>
5.1	Machine learning . . . . .	25
5.1.1	Supervised learning . . . . .	25
5.2	Artificial neural networks . . . . .	26
5.2.1	Activation functions . . . . .	28
5.2.2	Loss functions . . . . .	29
5.2.3	Gradient descent and backpropagation . . . . .	30
5.2.4	Optimization algorithms . . . . .	31
5.3	Convolutional neural network . . . . .	32
5.3.1	Pooling . . . . .	34
5.3.2	U-net . . . . .	35
<b>6</b>	<b>Papers</b>	<b>37</b>
6.1	Article I . . . . .	37
6.2	Article II . . . . .	52
6.3	Article III . . . . .	79

---

<b>7 Conclusion, discussion, and future work</b>	<b>117</b>
7.1 Key findings . . . . .	117
7.2 Discussion . . . . .	119
7.3 Future work . . . . .	120
<b>Bibliography</b>	<b>123</b>





# Chapter 1

## Introduction

### 1.1 Outline

The first chapter in the introduction will focus on the background, motivation, problem, and scope of the topic of this thesis.

The second chapter in the introduction focuses on the air-guns and how they produce bubble noise in marine seismic data. I explain why the bubble noise is a problem in seismic data and why it is important to remove it. Afterward, I mention some conventional methods of estimating and attenuating the bubble.

The third chapter in the introduction introduces the ghost and how it is produced in marine seismic data. I mention some of the problems with ghosts and why they usually are removed from seismic data. I describe some of the developed methods to remove ghosts.

The fourth chapter in the introduction introduces the basic concepts of migration and demigration. I focus on Kirchhoff depth migration and demigration in this chapter.

The fifth chapter in the introduction focuses on machine learning and artificial neural networks. Here, I mention some of the most important aspects of machine learning. Afterward, I describe the basics of artificial neural networks, convolutional neural networks, and the U-net I have used for debubbling and deghosting.

The sixth chapter summarizes each paper before I present the papers themselves.

The seventh chapter gives the key findings from each paper and a common context. Afterward, I discuss my work in relation to other research. Last, I discuss future research.

## 1.2 Background

Figure 1.1 shows that oil and gas are a major part of the world's energy consumption and have been since the 1950s. Even though renewable energy sources have increased in the last decades, oil and gas consumption has also increased [Ritchie et al., 2022]. Oil and gas will likely be a major part of future consumption until other energy sources can replace this demand. An energy transition could take decades, so it is important to find and extract more oil and gas until that happens.

Norway only produces around 2% of the global oil demand. However, in 2021, crude oil and natural gas amounted to 60% of the total value of Norway's exports of 2021 [Norwegian Petroleum, 2022]. Therefore, oil and gas export is vital to the Norwegian economy. Norway's (known) oil and gas reservoirs are located on the Norwegian Continental Shelf. To find oil or gas reservoirs, one usually needs geophysical methods to identify potential reservoirs before drilling a well. Depending on the drilling conditions, a well can easily cost 30 million USD [Oljedirektoratet, 2020]. Therefore, understanding the subsurface is crucial to saving money and resources. Seismic methods are one of the essential geophysical tools to image the subsurface. Seismic methods are used not only for finding oil but also to monitor gas and oil reservoirs or CO<sub>2</sub> storage.

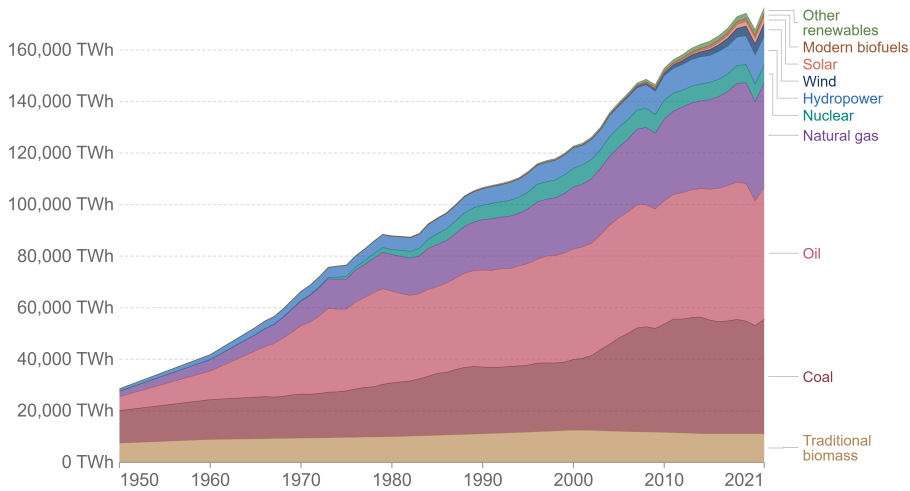


Figure 1.1: Global primary energy consumption by source from 1950 to 2021 [Ritchie et al., 2022].

Seismic methods are used on land and in water. A marine seismic acquisition usually consists of a vessel that drags a seismic source and streamers that contain hydrophones. Figure 1.2 illustrates a typical marine seismic acquisition. Seismic waves are emitted from a source that creates a displacement and pressure change in a medium. It is possible to detect seismic waves with receivers by measuring a change in displacement or pressure. A seismic survey uses a controlled source that releases seismic waves that travel through the earth and are later measured by receivers (Figure 1.2). The goal is generally to create an image of the subsurface or estimate the properties of the subsurface. However, noise contaminates the raw seismic data. Therefore, removing noise while preserving the signal is necessary after an acquisition. The end product is essential for seismic interpretation and inferring geological attributes from seismic inversion. Seismic interpretation and seismic analysis can help find valuable minerals, oil, gas, archaeological artifacts, monitor groundwater or glaciers, etc. Therefore, companies, institutions, and people are constantly working to create technology or methods that remove as much noise as possible while preserving the signal.

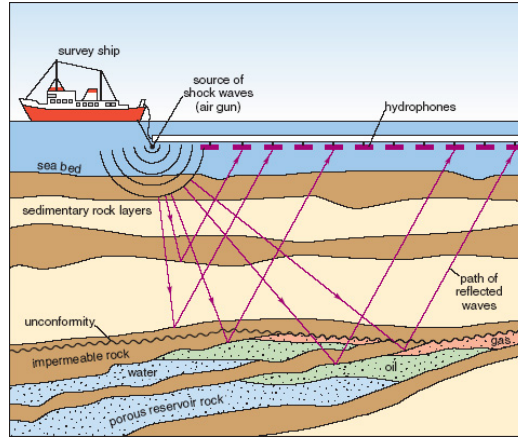


Figure 1.2: Illustration of a marine seismic acquisition [Learning Geology - A Geology Outreach Website, 2015].

In a marine seismic acquisition, the source usually consists of several air guns carefully positioned in an array. These air guns release over-pressured air into the water that expands quickly, creating a sharp seismic pulse. However, after the first expansion, the air bubble implodes and oscillates. Therefore, the first pulse will be accompanied by several smaller pulses. These secondary pulses are known as bubble noise.

Another problem during a seismic acquisition relates to the water-air contact at the water surface. The water-air reflection coefficient is close to  $-1$ . As a result, a seismic wave that propagates upwards towards the sea surface will be reflected downwards with almost the same amplitude and a polarity shift. This phenomenon is known as the seismic ghost. The ghost arrives at the receiver with a small delay compared to the primary.

We usually remove the bubble noise and ghost to obtain seismic data with less noise and a more accurate subsurface image. Several methods have been created to attenuate the bubble and ghosts. The bubble noise and ghost are described later in chapters 2 and 3.

### 1.3 Motivation

The recent availability of powerful GPUs, open-source software, and advances in artificial neural network structures have allowed neural networks to be applied to many geophysical problems. In seismic processing, neural networks have the potential to be applied

to many of the key processing steps (swell noise attenuation, interpolation, deghosting, debubbling, etc.). Conventional methods often require significant testing and computational power and can be sensitive to, e.g., noise. Neural networks are potentially robust to noise and different datasets. In addition, once trained, neural networks are computationally light and could save time. Using neural networks could change how we do seismic processing in the long term.

With these attractive properties, an artificial neural network could be helpful for debubbling and deghosting compared to traditional methods. Using neural networks for seismic processing is still an immature field. However, during the last few years, more research has been done within this field.

## 1.4 Problem

The challenge when using artificial neural networks is acquiring high-quality training data. Ideally, training data should be diverse and contain the complex features found in the prediction data. Using diverse training data could lead to a generalized network that produces good results on prediction data.

The problem with deghosting and debubbling is to create training data without the ghost or the bubble. Separation of the ghost or bubble noise is not perfect on real data. Therefore, ideal training data created from real data is not possible. However, training data created from real data contain the complex features found in the prediction data, which are desirable when training a network. Often, the only way to perfectly separate the ghost or bubble model is to create synthetic data. However, the main challenge with synthetic data is to model the complex features found in real data.

## 1.5 Scope

This thesis aims to use a convolutional neural network with a U-net structure to remove the ghost and the bubble noise from marine seismic data. In the first paper, I describe how a generalized network trained on real data can remove bubble noise over an extensive survey where the bubble noise changes. In the second paper, I use demigration to create training data that resemble real data. This training data can help a neural network to achieve good deghosting results on hydrophone-only data. In the third paper, we expand our method shown in the second paper to also deghost dual-component seismic data.



# Chapter 2

## Debubbling seismic data

### 2.1 Far-field source signature

In marine acquisitions, a seismic source often consists of an array of air guns. Air guns are desirable because they are reliable and with good signature repeatability. Each air gun releases compressed air into the surrounding water creating an acoustic pulse. A bubble of air expands and contracts, but the expansion is damped with each oscillation. Typically, the air bubble has a diameter (at standard operating depths of 5-10m) of about 1m [Ziolkowski, 1970]. At distances  $r$  of 1m or more, the pressure field of a single oscillating air-gun bubble can be written as [Ziolkowski and Johnston, 1997]:

$$p(r, t) - p_\infty = \frac{1}{r} s\left(t - \frac{r}{V_w}\right), \quad (2.1)$$

where  $p(r, t)$  is the water pressure at distance  $r$  and time  $t$ ,  $p_\infty$  is the hydrostatic pressure,  $V_w$  is the P-wave water velocity, and  $s(t)$  is the source function.

Figure 2.1 shows an example of a seismic response from an air gun in the far-field. The direct arrival is followed shortly by a surface reflection (ghost). However, after the ghost, we observe bubble pulses or bubble noise related to the air bubble oscillation. The signature is inconveniently long and oscillatory. Its frequency spectrum is multi-peaked and is not minimum phase [Ziolkowski et al., 1982]. The oscillation period of the air bubble changes with the depth, air gun volume, and pressure. By carefully designing an array of air guns with, e.g., different volumes, it is possible to attenuate the bubble noise recorded in the vertical far-field (Figure 2.2). However, one would still record residual bubble noise. This signature is more desirable because it is sharp, short, and with a



smooth and broad amplitude spectrum [Ziolkowski et al., 1982].

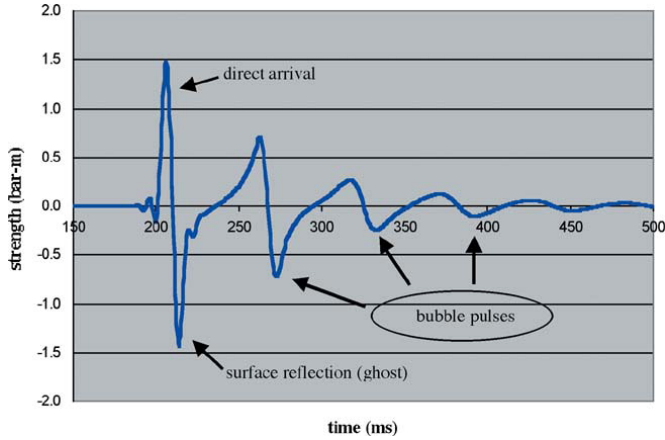


Figure 2.1: The vertical far-field signature of 40-in<sup>3</sup> air gun recorded by a hydrophone 300 m below the gun [Dragoet, 2000].

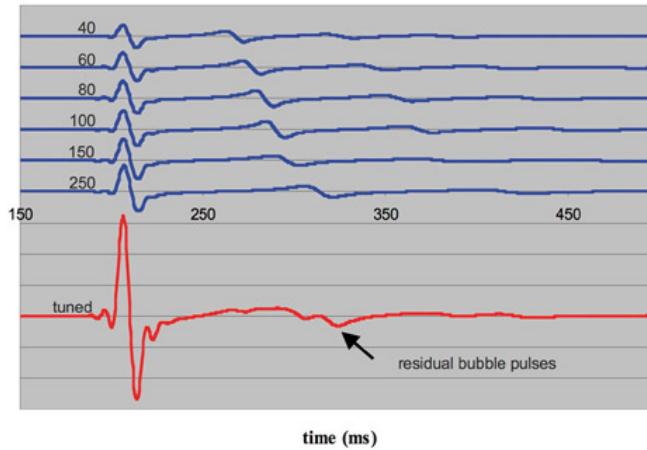


Figure 2.2: Far-field source signatures of individual air-guns shown with different gun volumes (blue). The red line show the combined far-field source signature if all air-guns are fired simultaneously [Landrø and Amundsen, 2010].

However, using an array of air guns has two consequences for the source signature. Firstly, the signature varies with departing direction of the acoustic waves from the source. Secondly, the signature pressure wave varies with distance from the source. The signature pressure wave becomes independent of distance in the "far-field" of the air-gun array at distances greater than  $D^2/\lambda$ , where  $D$  is the length of the array, and  $\lambda$  is the

wavelength of interest [Ziolkowski et al., 1982]. If we, e.g., have a 10 m array and are interested in frequencies up to 100 Hz in water, we need to be more than 7 m away from the array. Much work has been done to understand the bubble behavior and how to model far-field signature based on the air gun configuration [Johnson, 1994; Ziolkowski, 1970; Ziolkowski et al., 1982]. De-signature and de-bubbling are the main utilities of the far-field source signature.

One could place a receiver in the far-field to measure the vertical far-field source signature. However, this would usually not work on the continental shelf because sea bottom reflections contaminate the far-field signature. In addition, it is hard to know the true relative position of the array and receiver. Therefore, one normally estimates the far-field signature from indirect measurements or modeling techniques. If a survey has near-field hydrophones (NFH) measurements, it is possible to estimate the far-field signature from these measurements. Ziolkowski et al. [1982] argued that the pressure field in the water is simply the superposition of the spherical waves from all the bubbles, plus the reflections of these waves from the sea surface. The superposition of the direct and reflected waves from the  $n$  sources combined create the pressure at a point  $(x, y, z)$  in the water [Ziolkowski and Johnston, 1997]:

$$p(x, y, z, t) - p_\infty = \sum_{k=1}^n \left[ \frac{1}{r_k(t)} s_k \left( t - \frac{r_k(t)}{V_w} \right) - \frac{1}{R_k(t)} s_k \left( t - \frac{R_k(t)}{V_w} \right) \right], \quad (2.2)$$

where  $s_k(t)$  is the source time function of the  $k$ th source,  $r_k(t)$  is the distance from the  $k$ th source to the point  $(x, y, z)$ , and  $R_k(t)$  is the distance from the virtual bubble above the water surface to the point  $(x, y, z)$ .  $R_k(t)$  simulates the source ghost.  $r_k(t)$  and  $R_k(t)$  are time-dependent because the air bubble rises towards the sea surface.

Ziolkowski et al. [1982] suggested using  $n$  hydrophones as close as possible to the air guns but no closer than 1m. We then have the following equation:

$$p_i(x, y, z, t) - p_{i,\infty} = \sum_{k=1}^n \left[ \frac{1}{r_{ik}(t)} s_k \left( t - \frac{r_{ik}(t)}{V_w} \right) - \frac{1}{R_{ik}(t)} s_k \left( t - \frac{R_{ik}(t)}{V_w} \right) \right], \quad i = 1, 2, \dots, n, \quad (2.3)$$

where  $p_i(x, y, z, t) - p_{i,\infty}$  is the pressure field at the  $i$ th hydrophone,  $r_{ik}(t)$  is the distance from the  $k$ th source to the  $i$ th hydrophone, and  $R_{ik}(t)$  is the distance from the virtual bubble above the water surface to the  $i$ th hydrophone. We call the estimation of the source signatures at the hydrophones *notional* source signatures. Once the notional signatures are known, the pressure at any point  $(x, y, z)$  can be calculated using equation

2.2, including a point in the far-field.

Using NFH to estimate the far-field source signature is generally considered the most accurate method. However, when NFH measurements are unavailable, one can use other methods. One approach estimate the source signature from sea-bottom reflections that are aligned and stacked along one or more sail lines. However, this only works in deep water and on geology that change below the sea bottom. Otherwise, reflections below the sea bottom have constructive interference and contaminate the far-field estimate [Scholtz et al., 2015]. Another method uses the direct arrival to derive the far-field signature. Similarly to the technique using NFH, Davison and Poole [2015] use the direct arrival to estimate the notional source signatures using equation 2.3. This method is unsuitable in shallow water because the direct arrival quickly interferes with reflection data. Finally, it is possible to model the far-field signature based on the air-gun configuration [Landrø, 1992].

## 2.2 Debubbling

Despite using carefully designed air gun arrays, a significant amount of bubble noise is left in the far-field source signature. The bubble noise is generally unwanted in the recorded data and, therefore, removed. However, this requires an estimation of the far-field signature. de Jonge et al. [2022b] define the true far-field signature bubble noise as the *real bubble* and the estimate of the real bubble as the *bubble model* (Figure 2.3). The earth acts as a filter and is convolved with the source signature, meaning that the real bubble interferes with deeper reflections as shown in Figure 2.3c. Consequently, the bubble interferes with seismic reflections, which degrade the seismic image.

If the bubble model is a reasonable estimation, it is possible to find a 1D convolution operator to remove the real bubble from the seismic data. This operator is often called the debubbling filter and is equivalent to the inverse of the bubble model. To create the filter, one typically uses shaping or gapped deconvolution [Yilmaz, 2001]. The filter is applied to the seismic data (Figure 2.3c) to get debubbled seismic data (Figure 2.3e). Even if the real bubble change with direction (for an air-gun array), it is usually assumed that a 1D filter based on the vertical far-field real bubble is good enough for debubbling. de Jonge et al. [2022b] define the measured seismic data containing only the estimated bubble noise as the *bubble modeled data* (Figure 2.3d). The bubble modeled data is equivalent to the difference between the seismic data with the real bubble and the debubbled data.

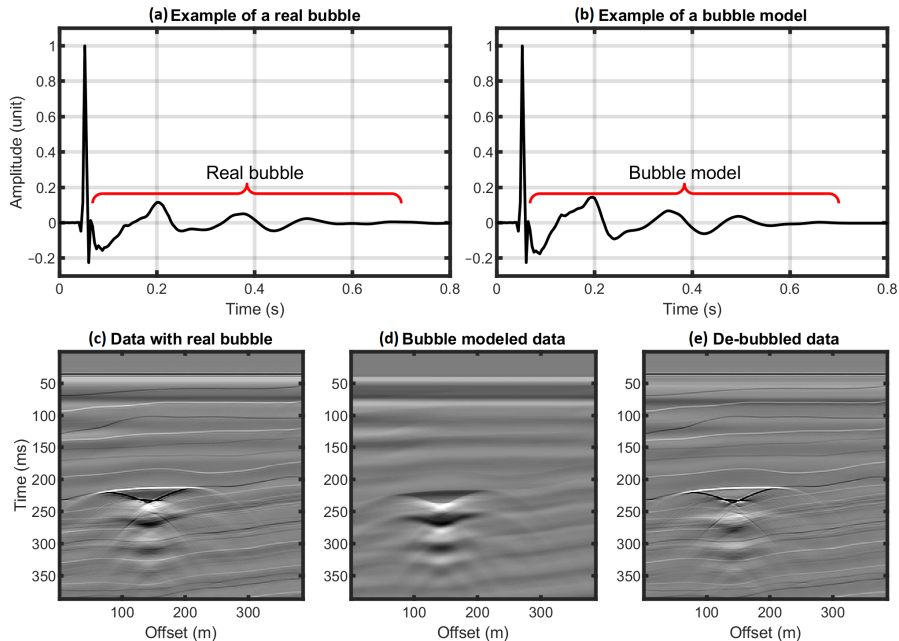


Figure 2.3: (a), (b) An example of a signature with a real bubble and bubble model, respectively. (c) Data with real bubble. (d) Bubble modeled data. (e) De-bubbled data. [de Jonge et al., 2022b].

Debubbling is an essential step in seismic processing and improves seismic data quality. However, if we derive a convolution operator from an incorrect estimate of the real bubble, the operator will leave residual bubble noise in the data, as shown in Figure 2.3e. Even if we know the source configuration, estimating the source signature can give different results for different methods. Figure 2.4 shows an example of this deviation where de Jonge et al. [2022b] have estimated the far-field source signature using modeling and NFH data. During a seismic acquisition, the air guns do not always release air with the same pressure from shot to shot. Occasionally, an air gun might not fire (drop-out), which changes the far-field signature significantly [Grion et al., 2018]. The physical conditions (such as weather, relative air-gun position, and the sea-surface reflection coefficient) can also change the far-field signature significantly within a survey [Ni et al., 2012]. The bubbles interact and are coupled with each other, meaning that a change in one bubble will also change the other. This interaction makes the system complex and can be hard to model correctly. Therefore, it is unrealistic that we can estimate the far-field signature perfectly and that the far-field signature does not change within a survey. In addition, calculating the far-field signature and designing a debubbling filter

for each shot is challenging and time-consuming.

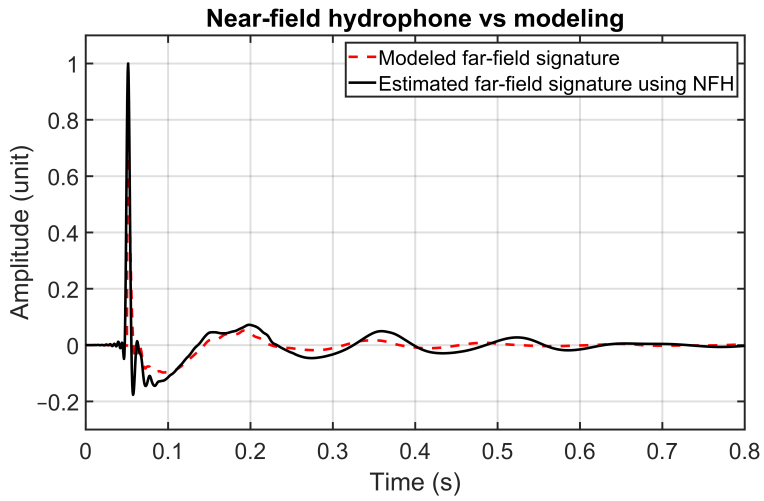


Figure 2.4: A far-field source signature estimated using NFH (black) or modeling (red) [de Jonge et al., 2022b].

# Chapter 3

## Deghosting seismic data

### 3.1 Seismic ghosts

A seismic source releases a seismic wavefield in all directions. Some of that energy travels upwards to the sea surface. The reflection of the up-going seismic wavefield at the sea surface is called the ghost. If we disregard the direct wave, the seismic wavefield can be reflected at the source and receiver sides, shown in Figure 3.1a. Therefore, a primary reflection is accompanied by three ghosts: the source ghost, the receiver ghost, and the source-receiver ghost. The sea surface's reflection coefficient depends on the incidence angle and frequency. However, in the frequency range of 0-100 Hz, the reflection coefficient,  $R$ , is usually close to -1 [Orji et al., 2013]. As a result, ghosts are near-total reflections with a polarity shift within this frequency range. Figure 3.1b illustrates these ghosts in a seismogram. The consequence of ghosts is an elongated and distorted seismic signature. The primary reflection and ghosts interfere constructively and destructively with different frequencies. Therefore, we observe peaks and notches in the frequency spectrum (Figure 3.1c). The notch frequency for pressure is given by [de Jonge et al., 2022c]:

$$f_n = n \frac{V_w}{2\Delta z \cos \alpha}, \quad n = 0, 1, \dots, \quad (3.1)$$

where  $V_w$  is the P-wave water velocity,  $\Delta z$  is the depth of the source or receiver, and  $\alpha$  is the incidence angle, respectively. The frequency notch will decrease if the receiver depth or source depth increase. However, the ghost notch at 0 Hz is present regardless of the source and receiver depths.

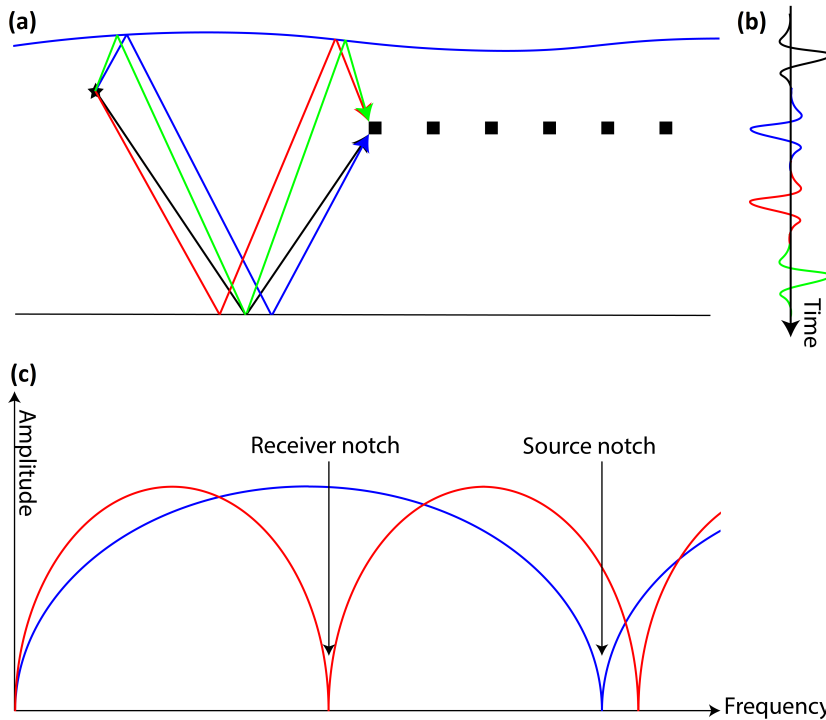


Figure 3.1: (a) Primary and ghosts arrivals shown as rays from the source to a receiver. (b) Primary and ghosts shown in the time domain for one receiver. (c) The ghost functions shown in the frequency spectrum for the source and receiver [de Jonge et al., 2022c].

Ghosts reduce the temporal resolution. Therefore, it is important to remove ghosts to improve the bandwidth and resolution. Consequently, deghosting could benefit geological interpretation [Hammond, 1962; Schneider et al., 1964]. Figure 3.2 shows an example of Marmousi finite-difference modeled data with and without ghosts. The source and receiver depths are 7m and 20m, respectively.

Historically, sources and receivers were located at relatively shallow depths (5-9 m) in conventional seismic acquisitions. Shallow sources and receivers result in a high-frequency ghost notch, which could limit the processing and imaging below the first notch. More recently, other acquisition methods aim to attenuate the receiver ghost. Slant streamers [Bearnth and Moore, 1989] and variable depth streamers [Rickett et al., 2014; Soubaras and Dowle, 2010; Soubaras et al., 2012] aim to diversify the ghost notch frequency along the streamer. Therefore, a CMP stack attenuates the receiver ghost. However, one typically attempt to deghost the data using processing techniques be-

fore a CMP stack. Over-under streamers is an acquisition method with cables towed in pairs at two different cable depths, with one cable vertically above the other [Hill et al., 2006]. Since the location of over/under towed streamers is at different depths, they also have a different ghost notch frequency. The seismic data recorded by the over/under towed streamers are combined into one dataset to attenuate the receiver ghost. Another acquisition method use multi-component streamers with pressure and particle velocity measurements to attenuate the receiver ghost [Carlson et al., 2007]. The receiver ghost has an opposite polarity compared to the primary for pressure. In contrast, particle velocity data have a primary and receiver ghost with the same polarity. When pressure and particle velocity data are combined, the receiver ghost is attenuated. Since the receiver ghost is attenuated, multi-component streamers can be towed deeper, enhancing the low frequencies. These acquisition configurations can help to attenuate the ghosts. However, they cannot alone solve the ghost problem.

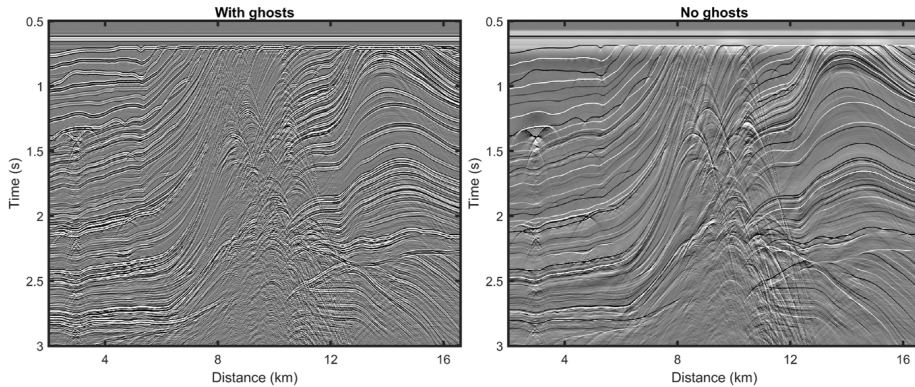


Figure 3.2: Example of Marmousi modeled data with and without ghosts.

## 3.2 Deghosting of pressure data

Deghosting of pressure data is still essential. Traditionally, one would estimate a deconvolution filter for source or receiver deghosting:

$$\frac{1}{G(f)} = \frac{1}{1 - e^{2i\pi 2\Delta z f/V_w}}, \quad (3.2)$$

where  $f$  is the frequency,  $G(f)$  is the ghost function, and  $V_w$  is the water velocity. Equation 3.2 assumes vertical ray paths. The simplest deghosting method uses the far-field source signature and deterministic deconvolution [Jovanovich et al., 1983], assuming



vertical ray paths. However, the ghosts change with the angle of propagation. As a result, the deconvolution filter in equation 3.2 is changed:

$$\frac{1}{G(f, k_x, k_y)} = \frac{1}{1 - e^{2i\pi 2\Delta z \sqrt{f^2/V_w^2 - k_x^2 - k_y^2}}}, \quad (3.3)$$

where  $k_x$  and  $k_y$  are the inline and crossline wavenumber.

Most receiver deghosting methods require common-shot gathers that are densely sampled in the inline direction. The reason is that the inline wavenumber is not related to the departing angle from the source but to the incidence angle for shot gathers. The relation between the departing and incidence angles is complex if the subsurface is complex. The energy emitted at a certain angle may arrive at one or multiple other angles. Figure 3.3 shows the frequency-wavenumber (FK) domain of a shot gather and receiver gather modeled using the Marmousi model [Martin et al., 2006]. We can observe the receiver ghost notch in the FK shot domain 3.3a, but it is not visible in the FK receiver domain 3.3b. In contrast, source deghosting methods are best suited for receiver gathers [Vrolijk and Blacquière, 2021]. However, the large gaps between shots limit the applicability of source deghosting in the common-receiver domain.

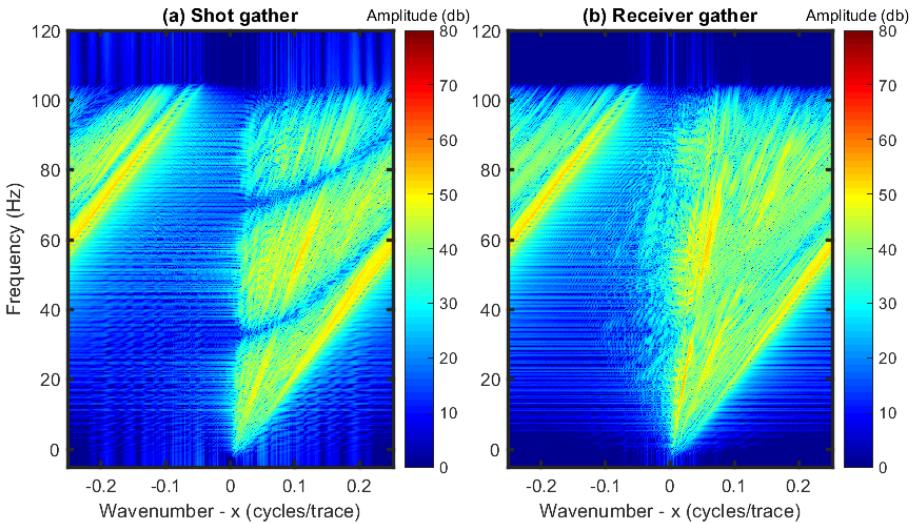


Figure 3.3: A shot gather (a) and receiver gather (b) shown in the FK domain.

Most current source and receiver deghosting methods aim to solve Equation 3.3. These deghosting techniques are typically applied in the FK domain Lindsey [1960], frequency-slowness domain Zhang et al. [2018],  $\tau - p$  domain [Poole, 2013; Wang et al., 2013],

or the space domain Amundsen et al. [2013a,b]. Deghosting is usually done pre-stack. However, deghosting can also be done post-stack, shown by Soubaras [2010].

### 3.3 Multi-component deghosting

Multi-component streamers have hydrophones and particle velocity sensors. Usually, the particle velocity sensors measure the vertical field (dual-component streamers). In some cases, multi-component streamers also measure the crossline field. Multi- or dual-component streamers are becoming more frequently used in the industry because vertical particle velocity ( $V_z$ ) data is beneficial for receiver deghosting. Therefore, these streamers can be towed deeper, which enhances the lower frequencies [Carlson et al., 2007].

If we define the positive measuring direction downwards, vertical particle velocity ( $V_z$ ) data have the same receiver ghost polarity compared to pressure data. However, pressure and  $V_z$  data have the opposite polarity of the up-going wavefield as shown in Figure 3.4. The receiver ghost notch frequency of particle velocity data is defined as [Carlson et al., 2007]:

$$f_n^V = \left| \frac{V_w}{2\Delta z \cos \alpha} \left( \frac{1}{2} + n \right) \right|, \quad n = 0, 1, \dots \quad (3.4)$$

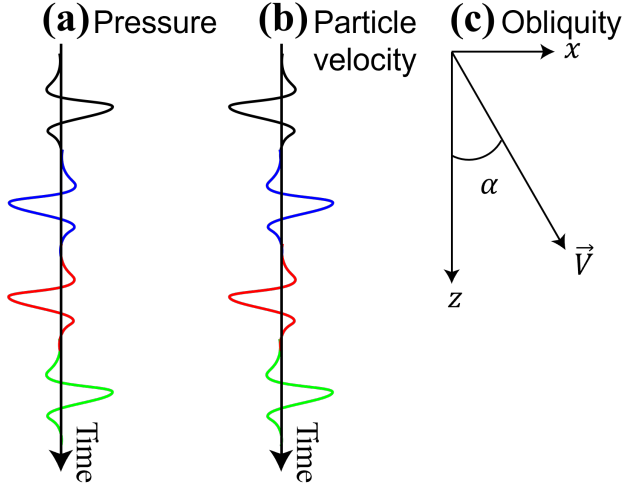


Figure 3.4: Illustration of two seismograms measured in pressure (a) and particle velocity (b) of a single reflection with the primary, source ghost, receiver ghost, and source-receiver ghost. (c) shows the obliquity angle ( $\alpha$ ) of the particle velocity ( $\vec{V}$ ) defined as positive downwards [Adopted from de Jonge et al. [2022a]].

The particle velocity receiver notch is located at the pressure receiver ghost peak, and vice versa. Therefore, adding the pressure and particle velocity data results in destructive interference of the receiver ghost. However, we must correct the obliquity before combining the pressure and  $V_z$  data. Obliquity is the deviation from the vertical and is defined by the angle,  $\alpha$ , shown in Figure 3.4. We can use the following equation when computing the pressure and  $V_z$  sum [Tenghamn and Dhelie, 2009]:

$$P_{UP} = \frac{1}{2}(P - FV_z), \quad (3.5)$$

where  $F$  is the obliquity filter:

$$F(\omega, k_z) = \frac{\rho\omega}{k_z} \quad \text{and} \quad k_z = \sqrt{\left(\frac{\omega}{V_w}\right)^2 - k_x^2}, \quad (3.6)$$

where  $k_z$  is the vertical wavenumber,  $\omega$  is the angular frequency, and  $\rho$  is the water density. The inline wavenumber,  $k_x$ , is usually sampled by receivers every 12.5 m along the seismic streamer. Figure 3.5 shows an example of a normalized obliquity filter and an FD-modeled shot gather from the Marmousi model in the frequency-wavenumber (FK)

amplitude domain. The obliquity filter in Figure 3.5 goes to infinity as we approach  $k_z = 0$ , meaning horizontally traveling waves. As a result, we cannot use the obliquity filter close to  $k_z = 0$  and have to mute or extrapolate the filter when we approach this value. Estimating  $k_z$  from  $k_x$  is a problem when we record aliased data. The pressure data in Figure 3.5 gets aliased as we go beyond the Nyquist frequency and observe wrap-around. The receiver spacing for this shot gather is 12.5 m. However, if we increase the receiver spacing or have data with higher frequencies, we observe more aliased data. Aliased data is a problem because it will cause a wrong scaling with the obliquity filter. As a result, the pressure and Vz sum does not fully remove the receiver ghost.

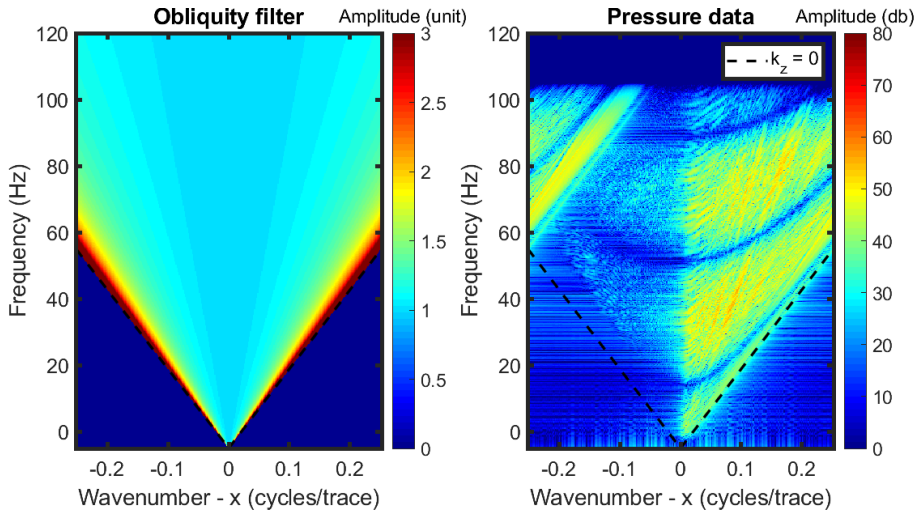


Figure 3.5: A normalized obliquity filter (left) and an example of clean Vz data in the FK amplitude domain (right). The black stippled line indicate when  $k_z = 0$  [de Jonge et al., 2022c]

A problem with Vz data is the strong noise measured during acquisition [Carlson et al., 2007; Tenghamn and Dhelie, 2009]. The Vz noise is mainly the result of longitudinal, transversal, or torsional streamer vibrations [Hlebnikov et al., 2021]. This Vz noise has a low frequency, primarily below 30 Hz. Figure 3.6a shows an example of real Vz noise. A seismic attenuation gain is used on this shot. Therefore, it appears that the Vz noise increase with depth, which it does not. The Vz data is often unusable for frequencies below  $\sim 20$  Hz [Mellier and Tellier, 2018; Peng et al., 2014; Poole and Cooper, 2018]. Therefore, we often rely on P-only deghosting for the low frequencies. However, we compute the P-Vz sum for the high frequencies. Figure 3.6b shows an example of Vz data with a low-cut filter. However, some methods still use pressure and Vz data on the low frequencies but avoid contamination of Vz noise. Poole and Cooper [2018] use a multi-

component inversion based receiver deghosting in the  $\tau - p$  domain. Telling and Grion [2022] is another example that uses inversion-based receiver deghosting with a hybrid operator in frequency and space. Both of these and most multi-component deghosting methods use weights to reduce the influence of Vz data on the low frequencies.

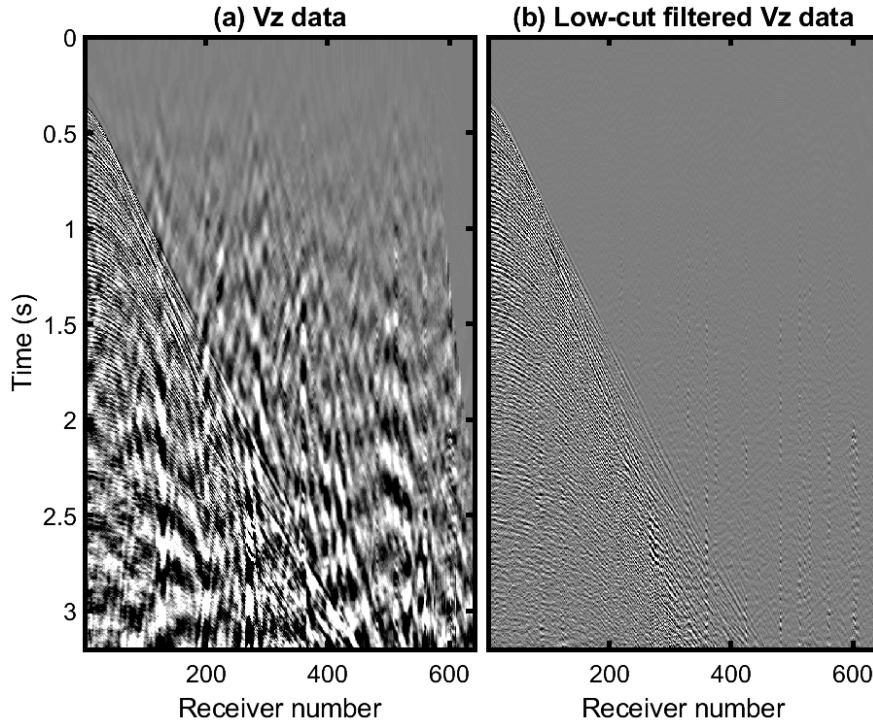


Figure 3.6: An example of Vz noise on a real shot gather (a) and the same shot gather with a low-cut (20 Hz) filter (b) [de Jonge et al., 2022c].

# Chapter 4

## Migration and demigration

### 4.1 Migration

Seismic migration aims to create an image of the subsurface reflectors from a seismic record. It can also be thought of as geometrically relocating recorded seismic events from where they originated in the subsurface [Santos et al., 2000a]. Migration is frequently used as an imaging technique and has been used for decades [Iversen et al., 2012]. Seismic migration can be divided into time migration and depth migration. Time migration creates an image in the time domain, while depth migration creates an image in the depth domain. Both are frequently used; however, most research is devoted to depth migration. Both time and depth migration requires a macro-velocity of the subsurface [Santos et al., 2000a]. Time migration has two advantages, it is a faster process, and the problem of estimating a velocity model for time migration is, in general, well-posed [Iversen et al., 2012]. However, pre-stack depth migration (PSDM) is the most accurate imaging method. Therefore, when the velocity model is complex, PSDM is usually the method of choice [Etgen and Kumar, 2012].

Migration can also be separated into pre-stack and post-stack migration. As the name suggests, post-stack migration is done after the stack, while pre-stack migration is done before the stack. Post-stack migration is much faster than pre-stack migration. However, pre-stack migration images are more accurate than post-stack migration images when we have a complex model with dips and lateral velocity variations. Therefore, pre-stack migration is usually preferred [Lines et al., 1994].

Kirchhoff PSDM is perhaps the most common migration method used today. This method is applied to data in the constant offset domain, the domain with source and

receiver pairs of a constant offset along a sail line. True-amplitude Kirchhoff constant offset migration integral for any migration domain point  $(\mathbf{x}, z)$  can be represented by [Bleistein, 1987; Santos et al., 2000b]:

$$IM(\mathbf{x}, z) = -\frac{1}{2\pi} \iint d^2\xi W_M(\xi, \mathbf{x}, z) \frac{\partial D(\xi, t)}{\partial t} \Big|_{t=\tau(\xi, \mathbf{x}, z)}, \quad (4.1)$$

where  $IM(\mathbf{x}, z)$  is the migrated data,  $W_M(\xi, \mathbf{x}, z)$  is the true-amplitude migration weight function, and  $D(\xi, t)$  is a trace in the data domain for a given surface location  $\xi = (\xi_x, \xi_y)$ . We integrate over the two-way traveltimes surface  $\tau(\xi, \mathbf{x}, z)$ . The two-way traveltimes is calculated with ray tracing from constant offset sources  $S(\xi)$  and receivers  $R(\xi)$  down to a fixed migration domain point  $(\mathbf{x}, z)$ . Figure 4.1 illustrates the migration of a constant traveltimes in the time domain to a diffraction point in the depth domain. Equation 4.1 is named true-amplitude because the AVO and AVA analysis can be done after the migration. The migration preserves the true-amplitude using the migration weight function,  $W_M$ , which varies along the traveltimes curve [Santos et al., 2000a].

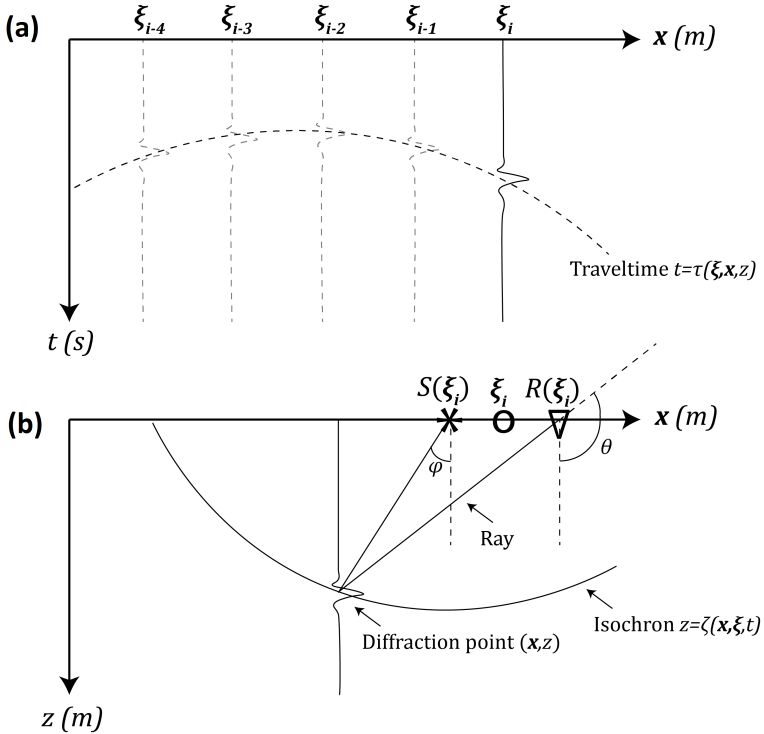


Figure 4.1: Illustration showing Kirchhoff migration and Kirchhoff demigration [de Jonge et al., 2022a]

## 4.2 Demigration

Seismic demigration is the reverse process of seismic migration. Therefore, true-amplitude Kirchhoff demigration involves a weighted mapping process from the depth-migrated image to the seismic data as shown in Figure 4.1. Consequently, the aim is to return to the un-migrated original common-offset section. The true-amplitude Kirchhoff constant offset demigration integral for any data domain point is  $(\boldsymbol{\xi}, t)$  is [Santos et al., 2000a; Tygel et al., 1996]:

$$P(\boldsymbol{\xi}, t) = \frac{1}{2\pi} \iint d^2\mathbf{x} W_D(\mathbf{x}, \boldsymbol{\xi}, t) \frac{\partial IM(\mathbf{x}, z)}{\partial z} \Big|_{z=\zeta(\mathbf{x}, \boldsymbol{\xi}, t)}, \quad (4.2)$$

where  $P(\boldsymbol{\xi}, t)$  is the demigrated data,  $IM(\mathbf{x}, z)$  is the migrated data,  $W_D(\mathbf{x}, \boldsymbol{\xi}, t)$  is the true-amplitude data weight function, and  $\zeta(\mathbf{x}, \boldsymbol{\xi}, t)$  is the isochron shown in Figure 4.1. The isochron surface,  $\zeta(\mathbf{x}, \boldsymbol{\xi}, t)$ , is calculated using ray tracing from a source-receiver pair to points with a constant time  $t$ . Equation 4.2 integrates over the isochron surface using a weighting function to preserve the true-amplitude.

Kirchhoff demigration and Kirchhoff modeling are closely related. The main difference is that Kirchhoff modeling uses a constructed model of layers (often based on the physical earth) with elastic parameters. However, Kirchhoff demigration uses a migrated image as the model.

Just like Kirchhoff migration, Kirchhoff demigration also requires a macro-velocity model. Demigration can use the same or another velocity model as used for migration. It is possible to demigrate a time-migrated image rather than a depth-migrated image. It is also possible to do demigration pre-stack or post-stack.

Kirchhoff demigration is useful in seismic interpretation, where it is beneficial to know what region on a reflector contributes to an observed primary reflection [Santos et al., 2000a]. Demigration is essential in my two papers about deghosting because I use demigration to create training data for the convolutional neural network. Demigration is an easy way of creating realistic training data that contain complex features similar to real data. Realistic training data is an advantage when training a neural network.





# Chapter 5

## Machine learning and artificial neural networks

### 5.1 Machine learning

Machine learning is a sub-field of artificial intelligence and covers multiple methods. These methods learn programs from data [Domingos, 2012]. In some cases, this is an attractive approach compared to "traditional" methods, where one manually has to construct the programs. Machine learning has spread rapidly in computer science, geoscience, and other fields in the last decade. Machine learning can be categorized into supervised, unsupervised, and reinforced learning. Reinforced learning tries to maximize a reward by taking action in an environment. Chess AI is an example of reinforced learning. Unsupervised learning aims to find hidden structures in a dataset. Supervised learning aims to build a mathematical model based on training data containing the inputs and desired outputs. From now on, I will focus on supervised learning.

#### 5.1.1 Supervised learning

Supervised learning methods need training data which is the input and output of the algorithm. The input is sometimes referred to as the feature, and the output is called the label. Supervised learning aims to find a function through iterative optimization of an objective function that maps the feature vectors (inputs) to labels (outputs) based on examples. After the function is found, it can map new examples.

Classification, active learning, regression, and structured output are popular types of

supervised learning. Classification algorithms are used when the output label is restricted to a few values. Active learning can interactively ask a user to provide labels for new data. Regression algorithms are used when the output can have any numerical value within a range. In this thesis, I work with structured output, which involves any task where the output is a vector or matrix with important relationships between the different elements [Goodfellow et al., 2016].

The fundamental goal of supervised learning is to generalize beyond the examples in the training dataset [Domingos, 2012]. Over-fitting is a common mistake in machine learning, where the success rate on training data creates an illusion of a successful mapping function. However, when the mapping function is introduced to unseen data, it can be as bad as random guessing. A validation dataset is often used to avoid over-fitting and to tune parameters. However, it is not used during training. The data is often divided into three parts, training data, validation data, and test data. Test data, sometimes called prediction data, is kept separate until the very end and is used as an unbiased evaluation of the mapping functions.

Acquiring good training data is essential if we want to generalize. A lousy algorithm with lots and good training data beats a clever algorithm with limited and flawed training data [Domingos, 2012]. However, time and resources limit how much training data we can use. In addition, methods require different computation times to learn from the same training data. Therefore, we should choose a machine learning method and training data based on the problem we want to solve.

## 5.2 Artificial neural networks

Deep learning and artificial neural networks are sub-fields of Machine Learning. The goal of a neural network is to estimate some unknown function  $\hat{f}$  such that we can map an input  $\mathbf{x}$  to an output  $\mathbf{y}$ . Feedforward networks are the most common types of artificial neural networks used today. A feedforward neural network is defined as  $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$  and learns the parameters  $\boldsymbol{\theta}$  that results in the best approximation of  $\hat{f}$  [Goodfellow et al., 2016]. It is called feedforward because information flows from the input  $\mathbf{x}$ , through the function  $f$ , to the output  $\mathbf{y}$ . Feedforward networks are important and widely used across multiple disciplines.

Neural networks consists of many functions  $f(\mathbf{x}) = f^{(n)}(f^{(n-1)} \dots (f^{(2)}(f^{(1)}(\mathbf{x}))) \dots)$ , where each function is referred to as a layer. The network's last layer is called the output layer,

and the layers between the input and output layers are called hidden layers. The number of hidden layers determines how "deep" the network is, and the name "deep learning" stems from this terminology. During training we want  $f(\mathbf{x})$  to match  $\hat{f}(\mathbf{x})$ . Training data provides us with both inputs  $\mathbf{x}$  and outputs  $\hat{\mathbf{y}} \approx \hat{f}(\mathbf{x})$ . Note that  $\hat{\mathbf{y}}$  is the training data, while  $\mathbf{y}$  is the output of the network. The learning algorithm must decide how to use each layer in the network to produce the desired outputs. Each layer is typically vector- or matrix-valued, and each layer's dimension determines the layers' width. We can think of each value in a layer as a unit that works in parallel with other units similar to neurons in a human brain.

Neural networks can obtain a non-linear learning algorithm which is an advantage for non-linear problems. The strategy of deep learning is to learn a non-linear transformation  $\phi$  to transform the input  $\mathbf{x}$  to a transformed input  $\phi(\mathbf{x})$ . The idea behind neural networks is that a simple feature can be learned from the data with one transformation. Complex features can be learned by combining many simple (non-linear) transformations.

I use an example network with a simple structure to explain each part of a feedforward neural network. The network contains three layers, with two units in the first two layers and one in the output layer. Figure 5.1a shows an illustration of this network. In this network, all the units between two consecutive layers are connected, which is called a dense neural network. Figure 5.1b shows the computations that occur from two units in the input layer to one unit in the second layer. Each feature is multiplied by a corresponding weight. Afterward, we take the sum of each product and feed it into a non-linear activation function. The output from the activation function is the input to a unit in the next layer. We can express the input to the hidden units as

$$g = h(\mathbf{W}^T \mathbf{x} + b), \quad (5.1)$$

where  $\mathbf{W}$  are weights (but only for a single layer in contrast to  $\boldsymbol{\theta}$ ),  $h(\cdot)$  is the activation function, and  $b$  is the bias. The bias is added to the product of features and weights. We use a bias to shift the activation function. We call the forward propagation the information flowing from the input  $\mathbf{x}$ , through the hidden units and producing  $\mathbf{y}$ . The output of this network is evaluated with a loss or cost function. Afterward, the loss or cost function gradient is used to update the weights using an optimization algorithm. Back-propagation allows the information from the cost function to flow backward through the network to compute the gradient [Goodfellow et al., 2016]. Each of these steps is explained in more detail later in this section.

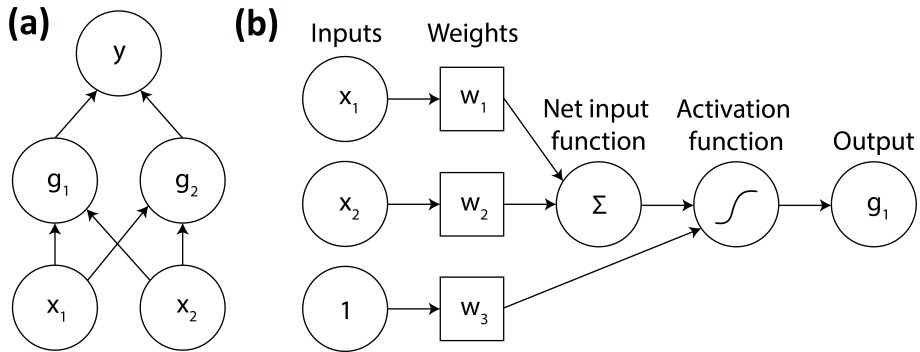


Figure 5.1: (a) Illustration of a simple dense neural network with three layers (Adopted from Goodfellow et al. [2016]). (b) Illustration showing the flow of information from two units in the first layer to one unit in the second layer.

### 5.2.1 Activation functions

There are many different activation functions. In modern neural networks, the default recommendation is to use the rectified linear unit (ReLU) [Goodfellow et al., 2016]. ReLU is defined as  $h(z) = \max\{0, z\}$ , where  $z = \mathbf{W}^T \mathbf{x} + b$ . Figure 5.2 shows some of the most used activation functions. The output units often use different activation functions compared to the hidden units. Examples of output activation functions are the identity (linear) function for a continuous output or the sigmoid function for a binary output.

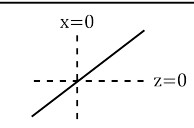
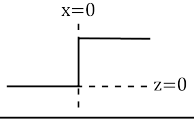
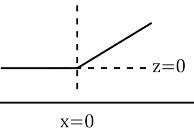
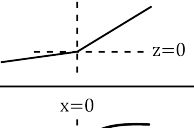
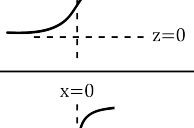
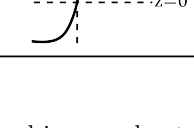
Activation function	Expression	Visualization
Identity	$h(z)=z$	
Binary	$h(z)=\begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$	
ReLU (Rectified linear unit)	$h(z)=\max\{0,z\}$	
Leaky ReLU	$h(z)=\begin{cases} \eta z & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$	
Sigmoid	$h(z)=\frac{1}{1+\exp(-z)}$	
Tanh (Hyperbolic tangent)	$h(z)=\frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$	

Figure 5.2: Activation functions that are often used in neural networks.

### 5.2.2 Loss functions

Depending on the problem, different loss functions are used. For a regression problem, the most common loss function is the mean squared loss (MSE):

$$L(f(\mathbf{x}; \boldsymbol{\theta}), \hat{\mathbf{y}}) = \frac{1}{2m} \sum_{i=1}^m (\hat{\mathbf{y}}^{(i)} - f(\mathbf{x}; \boldsymbol{\theta})^{(i)})^2. \quad (5.2)$$

The mean absolute loss (MAE) is also often used:

$$L(f(\mathbf{x}; \boldsymbol{\theta}), \hat{\mathbf{y}}) = \frac{1}{m} \sum_{i=1}^m |\hat{\mathbf{y}}^{(i)} - f(\mathbf{x}; \boldsymbol{\theta})^{(i)}|. \quad (5.3)$$

A more complex cost function, known as the structural similarity index (SSIM), is a metric that measures the similarity of two images [Wang et al., 2004]. In contrast, MSE and MAE assume that each pixel is independent.

For a classification problem, the log-loss function is often used:

$$L(f(\mathbf{x}; \boldsymbol{\theta}), \hat{y}) = \frac{1}{m} \sum_{i=1}^m \left( \hat{y}_i \log f(\mathbf{x}; \boldsymbol{\theta})^{(i)} + (1 - \hat{y}_i) \log(1 - f(\mathbf{x}; \boldsymbol{\theta})^{(i)}) \right). \quad (5.4)$$

Depending on the problem, these and other metrics can give different errors, influencing the gradient descent.

### 5.2.3 Gradient descent and backpropagation

The goal during training is to minimize (or maximize, depending on the metric) the loss by selecting appropriate values for the weights. We update the weights using gradient descent:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \gamma \nabla_{\boldsymbol{\theta}} L(\mathbf{x}_n, \hat{\mathbf{y}}_n), \quad (5.5)$$

where  $\gamma$  is the learning rate. The gradient descent converges towards a local minimum. To compute the gradient, we back-propagate the errors through the network. Backpropagation is an efficient way to calculate the gradient, which repeatedly uses the chain rule of calculus. If we use the following function as example:  $y = f(g(\mathbf{x})) = f(\mathbf{z})$  and  $\mathbf{z} = g(\mathbf{x})$ . The chain rule states that

$$\frac{dy}{dx_i} = \sum_j \frac{dy}{dz_j} \frac{dz_j}{dx_i}. \quad (5.6)$$

When using backpropagation, we want to estimate the gradient of the weights:

$$\nabla_{\boldsymbol{\theta}} L(\mathbf{x}_n, \hat{\mathbf{y}}_n) = \left( \frac{\partial L(\mathbf{x}_n, \hat{\mathbf{y}}_n)}{\partial w_1}, \dots, \frac{\partial L(\mathbf{x}_n, \hat{\mathbf{y}}_n)}{\partial w_k} \right), \quad (5.7)$$

where  $k$  is the number of weights. Figure 5.3 show how back-propagation is used to update the weights of an example network. In this example, the partial derivatives  $\partial L_{w_4}$  and  $\partial L_{w_5}$  are related to the weights from the last hidden layer to the output layer. Usually, there are multiple hidden layers with multiple units in each layer, which quickly

equates to many weights that need updating.

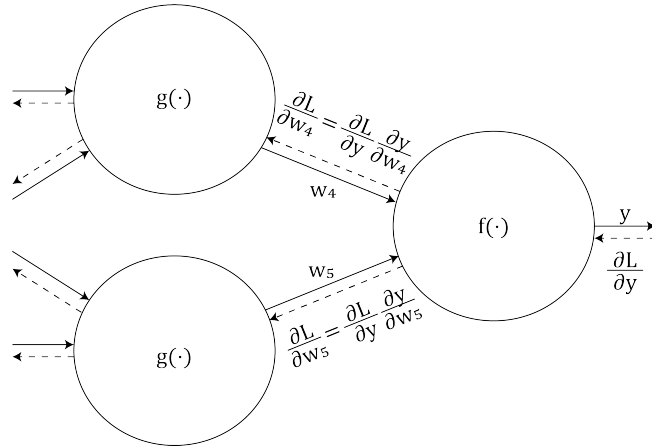


Figure 5.3: Illustration of how the chain-rule and back-propagation is used to update weights.

### 5.2.4 Optimization algorithms

Optimization algorithms have the goal of finding the weights  $\theta$  of a neural network that reduces a loss function  $L(\mathcal{D})$ , where  $\mathcal{D}$  is a dataset containing training data [Goodfellow et al., 2016]. Neural network optimization algorithms differ from traditional optimization algorithms mainly because when training neural networks, the goal is not necessarily to find the minimum of a loss function but to reduce the loss function in the hope that it will improve the network performance on a test dataset. However, in traditional optimization algorithms, the main goal is to find the minimum of a loss function. Neural network optimization algorithms do not usually stop based on a loss function criterion on the training dataset but on a validation dataset.

Computing the gradient of the loss for every training example  $\nabla_{\theta} L(\mathbf{x}_n, \hat{\mathbf{y}}_n)$  can be expensive. Therefore, one usually takes the average loss to batches of random examples before the gradient is calculated. As a result, the approximate (noisy) gradient is calculated quickly. Small batches can offer regularization effects, perhaps because of the noise they bring to the calculation of the gradient.

The most basic optimization algorithm is the gradient descent method in Equation 5.5. However, multiple other optimization algorithms are better behaved and can accelerate the training process. However, the most used optimization algorithm for neural net-



works is probably Stochastic Gradient Descent (SGD) and its variants. In contrast to Gradient Descent, SGD uses mini-batches to compute the average gradient:

$$\nabla_{\theta} L = \sum_{i=1}^l \nabla_{\theta} L(\mathbf{x}_i, \hat{\mathbf{y}}_i), \quad (5.8)$$

where  $l$  is the size of the minibatch. The minibatch size is usually small and chosen to be a power of 2 (because that will make GPUs run more efficiently). One epoch is one pass through the whole training data  $N$ , meaning we estimate the gradient  $N/l$  number of times. It is normal to run multiple epochs before training is complete.

Different adjustments to the SGD algorithm use momentum or adaptive learning rate. Momentum uses the current gradient and the sum of previous gradients to update parameters. Adaptive learning adjusts the learning rate during training. Examples of adaptive learning algorithms are AdaGrad, RMSProp, and Adam [Goodfellow et al., 2016].

### 5.3 Convolutional neural network

Convolutional neural networks (CNNs) are commonly used on image data. CNNs use convolution in place of general matrix multiplication in at least one of their layers [Goodfellow et al., 2016]. Convolution of two discrete 1D functions  $x$  and  $w$  is given by:

$$s(t) = (x * w)(t) = \sum_{\alpha=-\infty}^{\infty} x[\alpha]w[t - \alpha]. \quad (5.9)$$

In CNNs,  $x$  is referred to as the input,  $w$  as the kernel, and  $s$  sometimes as the feature map. Equation 5.9 can be expanded to include 2D, 3D, or any higher dimensional functions. During training, the goal is to learn the weights of the kernels. Figure 5.4 illustrates a 2D convolution of a  $3 \times 4$  input and  $2 \times 2$  kernel.

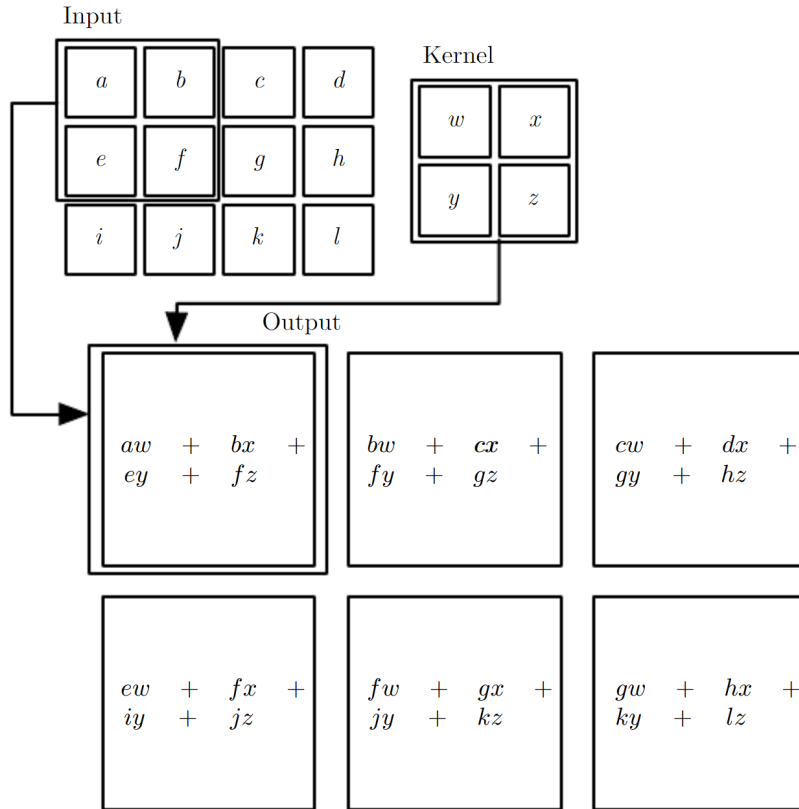


Figure 5.4: Illustration of a 2D convolution [Goodfellow et al., 2016].

CNNs have sparse interactions between input and output units in contrast to matrix multiplication, where all input and output units interact. However, the sparsity depends on the kernel size, and CNNs have sparse interactions because the kernel is smaller than the input. Figure 5.5 shows a network with sparse interactions and a fully connected network. Figure 5.5 also shows that units in deeper layers can interact with a large portion of the input in a deep CNN. CNNs assume meaningful features are closely connected within tens or hundreds of pixels. Therefore, CNNs store fewer parameters, reducing the memory requirement and computational cost.

Another difference between CNNs and traditional neural networks is that the kernel weights in a CNN can be used more than once [Goodfellow et al., 2016]. In contrast, weights in a traditional network are used exactly once when computing the layer's output. In a CNN, each kernel weight is used at every input position (sometimes excluding boundary pixels). As a result, we only need to learn one set of parameters for all locations

rather than one set of parameters at each location. CNNs can, therefore, significantly reduce the memory requirement.

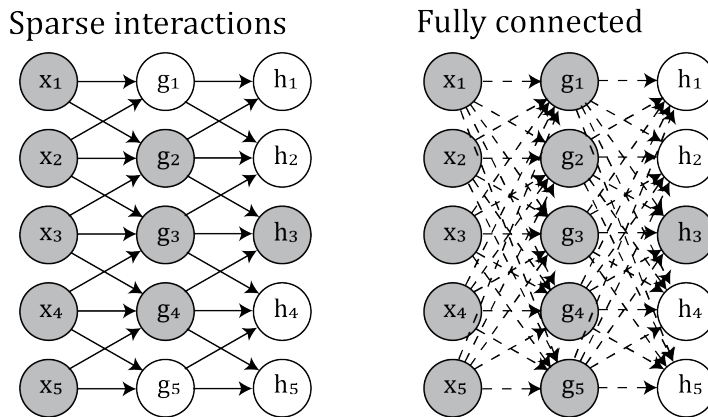


Figure 5.5: Showing a CNN network with sparse interactions (left) and a dense network with fully connected units (right). The arrows indicate which units interact with each other. The  $h_3$  unit in the last layer receives information from the gray-colored units.

An attractive feature of CNNs is that layers have a property called equivariance to translation [Goodfellow et al., 2016]. If the input changes, the output will change in the same way. For example, if we shift the input image, the output will also be shifted with the same representation compared to the output of a non-shifted image.

### 5.3.1 Pooling

Convolutional layers often consist of three stages: the convolution stage, then an activation function is used, and in the last stage, we use a pooling function [Goodfellow et al., 2016]. A pooling function reduces the dimensions of the data by computing a summary statistic of nearby outputs into a single unit. One of the most popular pooling functions is max pooling, which calculates the maximum output within a rectangular neighborhood. Pooling functions such as the  $L^2$  norm, average, or weighted average are also used. Pooling functions help make the network invariant to small translations of the input. In other words, if we translate the input by a small amount, the output is similar to the output of a non-translated input. Invariant to small translations are helpful when we care more if a feature is present rather than the exact location of a feature. Pooling function can also make the network more efficient because it reduces the dimension.

### 5.3.2 U-net

U-net is a CNN structure first introduced by Ronneberger et al. [2015]. The name of this network is inspired by the u-shaped architecture that uses upsampling and downsampling. This structure is similar to an encoder-decoder network typically used for dimension reduction. However, the U-net use skip connections which copy feature maps from one layer to another. Skip connections are used to maintain high-resolution features from the contracting path to the expansive path. The encoder part of the network downsamples the number of pixels, while the decoder part upsamples the number of pixels. Pooling functions can be used for downsampling. In contrast, transposed convolution upsample the number of pixels, which is an operation that goes in the opposite direction of a convolution [Dumoulin and Visin, 2016]. Figure 5.6 depicts an example of convolution and transposed convolution. In this example, the convolution of a  $4 \times 4$  input with a  $3 \times 3$  kernel with unitary stride and no padding results in a  $2 \times 2$  output. Therefore, the transposed convolution could be calculated with a convolution of a  $2 \times 2$  input on a  $3 \times 3$  kernel with a unitary stride and  $2 \times 2$  zero padding. The stride is the distance between two consecutive positions of the kernel. This network has a structured output, such as an image, which makes it suitable for image segmentation, image processing, etc. Figure 5.7 shows the U-net by Ronneberger et al. [2015], which they used for biomedical segmentation problems. However, the U-net has been used in various disciplines, including seismic processing [de Jonge et al., 2022b,c; Hlebnikov et al., 2022; Vrolijk and Blacquière, 2021]. Since the U-net uses downsampling to reduce the image size, it is more efficient than a CNN with no downsampling. In addition, the downsampling increases the network's receptive field, which can be an advantage [Lucas et al., 2018].

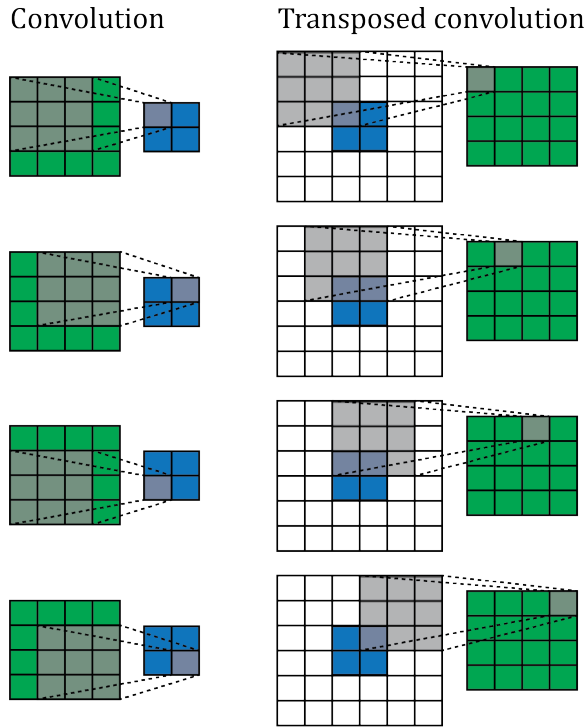


Figure 5.6: Depiction of convolution and transposed convolution.

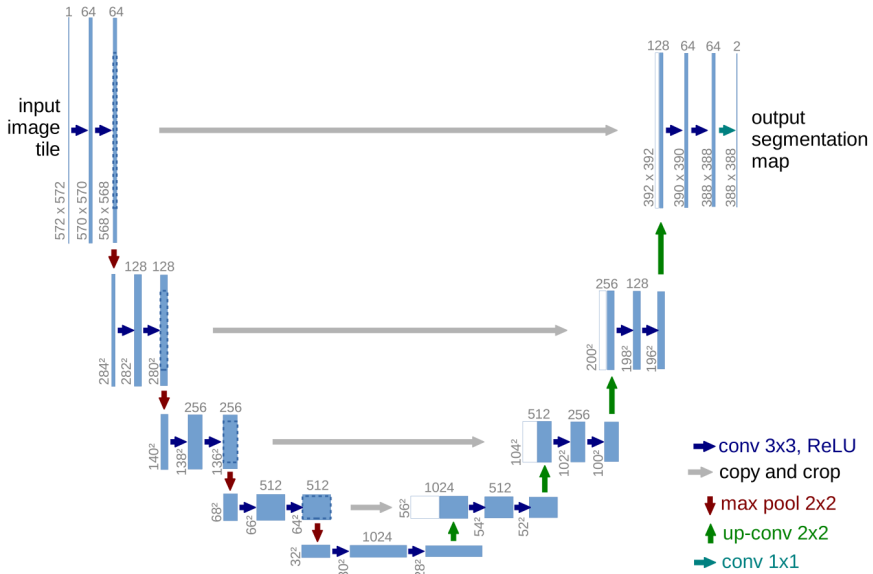


Figure 5.7: The U-net structure used by Ronneberger et al. [2015].

# Chapter 6

## Papers

In this chapter, I present the three papers that make up the central part of this thesis. All my research is presented in these three papers. Papers 1 and 2 are published and have undergone peer review. Paper 3 has been submitted to review. The papers have deviating page numbers from the rest of the thesis. In addition, each paper has its own reference style and references that are independent of the thesis.

### 6.1 Article I

Title: *Debubbling seismic data using a generalized neural network*

Authors: Thomas de Jonge, Vetle Vinje, Gordon Poole, Song Hou, and Einar Iversen

Published in *Geophysics* 87, 1, V1-V14 (2022). DOI: 10.1190/GEO2021-0053.1

Published version reprinted with permission from SEG. Copyright: Society of Exploration Geophysicists 2022.

## 6.2 Article II

Title: *Source and receiver deghosting by demigration-based supervised learning*

Authors: Thomas de Jonge, Vetle Vinje, Peng Zhao, Gordon Poole, and Einar Iversen

Published online in *Geophysical Prospecting*, 1-26 (2022). DOI: 10.1111/1365-2478.13253

Published version reprinted. Published under Open Access license CC-BY 4.0. Copyright: The Authors 2022.

Received: 23 February 2022 | Accepted: 12 July 2022

DOI: 10.1111/1365-2478.13253

ORIGINAL ARTICLE

Geophysical Prospecting



# Source and receiver deghosting by demigration-based supervised learning

Thomas de Jonge<sup>1,2</sup> | Vetle Vinje<sup>2</sup> | Peng Zhao<sup>2</sup> | Gordon Poole<sup>3</sup> | Einar Iversen<sup>4</sup>

<sup>1</sup>Department of Earth Science, University of Bergen, Bergen, Norway

<sup>2</sup>CGG Services Norway AS, Lilleaker, Norway

<sup>3</sup>CGG Services, R&D Crompton Way, West Sussex, UK

<sup>4</sup>Department of Earth Science, University of Bergen, Bergen, Norway

## Correspondence

Thomas de Jonge, Department of Earth Science, University of Bergen, Allégaten 41, Bergen, 5007, Norway.  
E-mail: thomas.jonge@uib.no;  
thomas.dejonge@cgg.com

## Funding information

Research Council of Norway, the University of Bergen and CGG for funding this work through an industrial Ph.D., Grant/Award Number: 305450

## Abstract

Deghosting of marine seismic data is an important and challenging step in the seismic processing flow. We describe a novel approach to train a supervised convolutional neural network to perform joint source and receiver deghosting of single-component (hydrophone) data. The training dataset is generated by demigration of stacked depth migrated images into shot gathers with and without ghosts using the actual source and receiver locations from a real survey. To create demigrated data with ghosts, we need an estimate of the depth of the sources and receivers and the reflectivity of the sea surface. In the training process, we systematically perturbed these parameters to create variability in the ghost timing and amplitude and show that this makes the convolutional neural network more robust to variability in source/receiver depth, swells and sea surface reflectivity. We tested the new method on the Marmousi synthetic data and real North Sea field data and show that, in some respects, it performs better than a standard deterministic deghosting method based on least-squares inversion in the  $\tau$ - $p$  domain. On the synthetic data, we also demonstrate the robustness of the new method to variations in swells and sea-surface reflectivity.

## KEYWORDS

data processing, modelling, noise, signal processing, seismics

## INTRODUCTION

Ghosts are the result of a reflection of the up-going seismic wavefield at the sea surface on the source and receiver side as shown in Figure 1 (we do not consider the direct wave in this research). A consequence of this ghost reflection is that a reflection from a subsurface structure (black ray in Figure 1a) will be followed by a source ghost, a receiver ghost and a combined source–receiver ghost that elongates and distort the seismic signature. These events interfere constructively and destructively at different frequencies and create a complex recorded trace (Figure 1b). As a result, we observe peaks and notches in the frequency spectrum (Figure 1c). Ghost notches within the seismic bandwidth are problematic because they

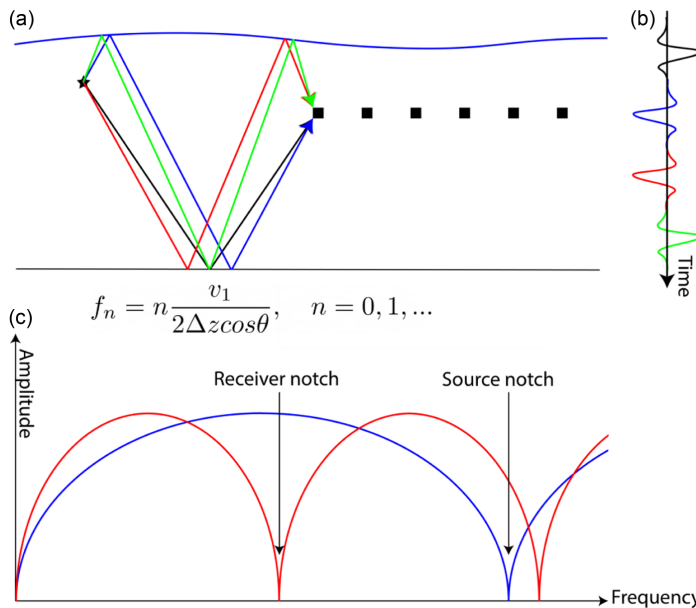
attenuate some frequencies and reduce the temporal resolution (Carlson et al., 2007; Poole, 2013). Removing ghosts improves the bandwidth, resolution and signal-to-noise-ratio of the seismic data, bringing benefits for seismic inversion and geological interpretation (Song et al., 2015). Referring to Figure 1a, a successful source and receiver deghosting would imply that up-going energy on the source side (green and blue rays) and down-going energy on the receiver side (red and green rays) will be removed. In other words, we remove the source, receiver and source–receiver ghosts (blue, red and green rays).

The frequency and amplitude of the ghost notches are dependent on the reflectivity and incidence angle at the water surface, the depth of the source and receivers and the water

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *Geophysical Prospecting* published by John Wiley & Sons Ltd on behalf of European Association of Geoscientists & Engineers.





**FIGURE 1** (a) Ray paths from a source (star) to receivers (squares) of the primary (black), source ghost (blue), receiver ghost (red) and source–receiver ghost (green). (b) The primary and ghosts in the time domain for one receiver. (c) Ghost functions for the source (blue) and receiver (red). The ghost notch frequency formula (Aytun, 1999) is shown in Figure 1c, where  $v_1$ ,  $\Delta z$ , and  $\theta$  are the water velocity, source/receiver depth, and incidence angle, respectively.

velocity. With an increase in source or receiver depth, there is a decrease in the notch frequency as shown in Figure 1c and the following equation (Aytun, 1999):

$$f_n = n \frac{v_1}{2 \Delta z \cos(\theta)}, \quad n = 0, 1, \dots, \quad (1)$$

where  $v_1$ ,  $\Delta z$  and  $\theta$  are the water velocity, source/receiver depth and incidence angle, respectively. The ghost notch at 0 Hz follows this rule when  $n = 0$  and is present regardless of the source and receiver depth. Removing the 0 Hz notch will recover valuable low-frequency information.

Historically, the conventional seismic acquisition solution to the deghosting problem was to locate sources and receivers at relatively shallow depths, typically 5–9 m and to limit the processing and imaging to the frequencies below the first notch. The deghosting was done by deterministic deconvolution (Jovanovich et al., 1983) with the assumption of vertical ray paths and a best-guess sea surface reflectivity. Although other approaches, such as slant streamers (Bearth & Moore, 1989) or over-under streamers (Hill et al., 2006), were proposed, shallow sources and receivers dominated until the recent advent of broadband seismic acquisition. First, a multi-component streamer approach was proposed

by Carlson et al. (2007), which allowed the streamers to be towed deeper, enhancing the low frequencies. Later, variable-depth streamer acquisition and processing solutions were also proposed (Soubaras & Dowle, 2010; Soubaras et al., 2012; Rickett et al., 2014).

These new configurations in acquisition help to attenuate the receiver side ghost and illustrate the benefits of broadband seismic. However, changes in acquisition geometry alone cannot solve the deghosting problem. For multi-sensor acquisition, the particle velocity measurements are typically noisy and unusable for frequencies below 15–20 Hz (Peng et al., 2014; Mellier & Tellier, 2018; Poole & Cooper, 2018). For variable-depth acquisition, notch diversity dilutes the impact of the receiver ghost over a range of frequencies, but a processing solution is still necessary to remove the receiver ghost wavefield. In addition, both acquisition methods suppress only the receiver side ghost, leaving the source side ghost untouched.

Consequently, source and receiver side deghosting of single-sensor (hydrophone) seismic data is still important. The simplest single-sensor deghosting method consists of deterministic deconvolution as mentioned above. In the past decades, more advanced methods have been introduced. Aytun (1999) modelled a flat streamer receiver ghost and

removed the ghost in the fourier-wavenumber ( $f$ - $k$ ) domain. Soubaras (2010) used joint deconvolution of migration and a mirror migration. Amundsen et al. (2013b, 2013a) perform a space-domain deghosting based on a Green's function. Amundsen and Zhou (2013) presented a source and receiver deghosting method using the inverse Fourier transform. Furthermore, frequency-slowness domain inversion for deghosting was described by Zhang et al. (2018). In addition, Poole (2013) and King and Poole (2015) performed a  $\tau$ - $p$  domain inversion for deghosting.

Recently, multiple papers using neural networks in seismic processing have been published. Examples of seismic processing applications of a network are interpolation (Greiner et al., 2019; Fang et al., 2021), denoising (Klochikhina et al., 2020), seismic interference noise attenuation (Sun et al., 2019) and debubbling (de Jonge et al., 2021). An advantage of a neural network is its ability to recognize patterns in the data and adapt to changing patterns. An example of this is given by de Jonge et al. (2021), who showed how a generalized network can debubble data when the source signature changes from shot to shot.

We have found three papers using machine learning for the application of deghosting. Vrolijk and Blacquiere (2020) use a neural network for source deghosting in the common receiver domain. They use conventional receiver deghosting in the common shot domain to create training data and later apply a trained network in the common receiver domain for source deghosting. A drawback with this approach is that the quality of the conventional receiver deghosting used in the training could limit its accuracy. Almuteri and Sava (2021) use a network for deghosting trained on the Marmousi model and Sigsbee model and tested on the Amoco statics test model. Their method needed the real data acquisition geometry and seafloor bathymetry to create training data. Peng et al. (2021) demonstrated a new network structure called DUNet on a deghosting example. However, they use conventional deghosting on part of the survey to create training data with and 'without' ghosts and apply the trained network to the rest of the survey.

A challenge with supervised neural networks is acquiring training data that contain complex features similar to the real data. Some papers show how to use pure synthetic data to train a network that is later used on real data (Zu et al., 2020; Qu et al., 2021). Other papers show how to train a network on synthetic data and use real data for fine-tuning (Cunha et al., 2020; Li et al., 2020). Another option is to use conventional deghosting methods to create training data (Peng et al., 2021). In addition, some papers utilize similarities between training and inference datasets in two different domains, respectively (Siahkoohi et al., 2018; Greiner et al., 2019; Vrolijk & Blacquiere, 2021).

In this paper, we propose a new approach for source and receiver deghosting using demigration-based supervised

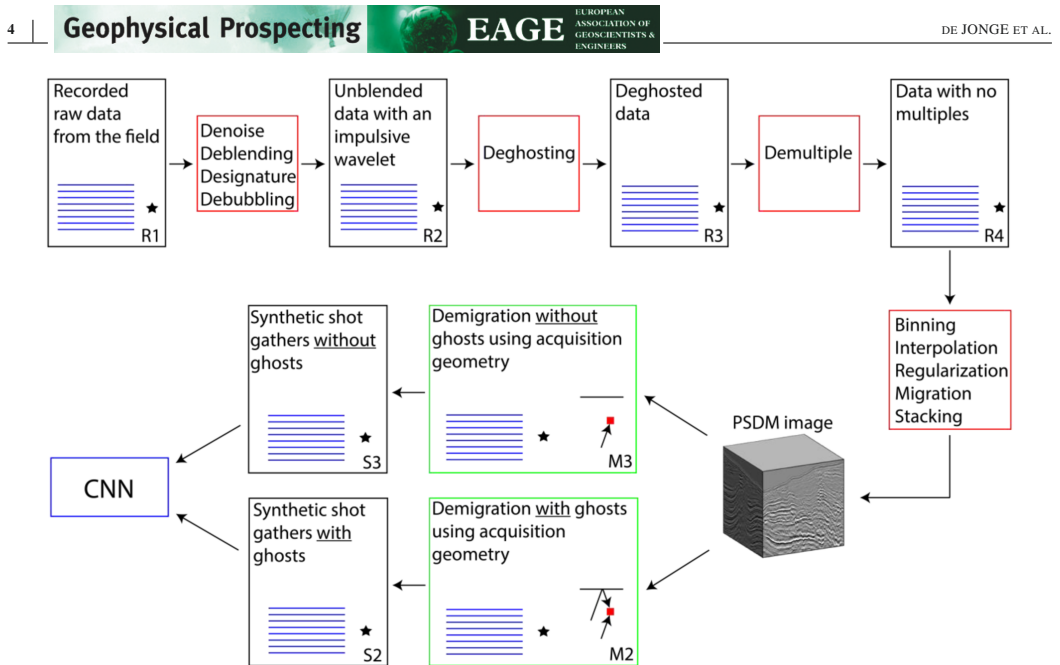
learning for hydrophone-only seismic data. We use the acronym DEGDEM (DEGghosting using DEMigration-based supervised learning) for this method. Training data are modelled by Kirchhoff demigration (Santos et al., 2000a) from a pre-stack depth migration (PSDM) image creating a set of ghosted and non-ghosted shot gathers with the real source/receiver geometry for the training. The demigrated shot gathers are termed the *synthesized data* (as opposed to synthetic data) as they are generated from PSDM images from real recorded data. There are several advantages of DEGDEM. First, we do not need to build a detailed P-wave velocity and density model for the synthetic modelling. Instead, a smooth velocity model is used, which is also used in the migration and is easily available. Second, for a given velocity model, the inverse of true-amplitude migration is true-amplitude demigration (Santos et al., 2000b) so the training data will be similar to the real data.

The proposed method is used on real data from the North Sea, Tampen area (CGG, 2020). In addition, we tested DEGDEM on synthetic finite-difference data using the Marmousi model (Martin et al., 2006). The tests on the Marmousi model allowed us to quantify the effect of training approaches to make the network adaptable to changes in receiver depth and the reflection of the sea surface. It also allowed us to quantify the sensitivity to residual ghosts in the PSDM image. We also test DEGDEM on multiples with ghosts. This test is important because the training data from the demigration does not contain multiples. Results on both synthetic and real data are compared with a conventional deghosting method by Poole (2013).

## METHODOLOGY

### Training, validation and test data

A challenge with supervised neural networks is to find or create training data – with and without ghosts – that is similar to the real data. Ideally, using real data as training data would be the preferred option. However, obtaining marine seismic data without ghosts is not straightforward. One option is to use an existing hydrophone-only deghosting method to attenuate the ghost and use data before and after deghosting for training (Peng et al., 2021). The main problem with this approach is that the network is trained on data that are not perfectly deghosted. For that reason, it is hard for the network to achieve better results than conventional deghosting. A second option is to use deghosting from multi-sensor streamer acquisitions. This would involve training a network to produce the sum of pressure and vertical velocity data from hydrophone data and using this network for a nearby hydrophone-only dataset. Towed streamer accelerometer data are not always available, and the accelerometer recordings



**FIGURE 2** An illustration of the DEGDEM workflow used in this paper. Several processing steps remove unwanted energy and are followed by migration to obtain a stacked PSDM image followed by demigration to generate synthesized shot gathers for the training.

are often noisy, particularly at low frequencies. Another option is to use synthetic data generated in a velocity–density model using, for example, finite-difference modelling. The challenge with this approach is building a velocity–density model, which will lead to finite-difference data mimicking the spectral content, phase shift and event complexity of the real data.

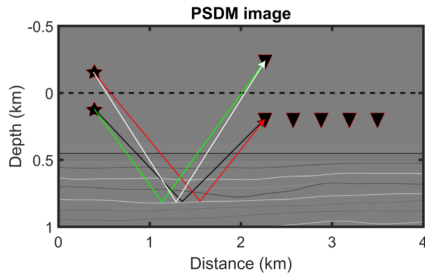
The new approach, DEGDEM (DEGhosting using DEMigration-based supervised learning), described in this paper involves generating *synthesized* shot gathers by Kirchhoff demigration (Hubral et al., 1996; Santos et al., 2000a) from a seismic image. The seismic image is a representation of the true subsurface reflectivity and is created from seismic data using Kirchhoff migration. As Kirchhoff migration and demigration can be regarded as inverse processes (Hubral et al., 1996; Santos et al., 2000b) or adjoint processes (Schuster, 1993), the synthesized seismic data will closely resemble the recorded seismic data. The demigration approach is described in more detail in the subsection ‘Kirchhoff demigration’.

The DEGDEM workflow is illustrated in Figure 2. As part of a standard processing flow, recorded data proceed through several processing steps including denoise, deblending, designature, debubbling, deghosting and demultiple. This creates shot gathers with (ideally) an impulsive wavelet, no ghosts and no multiples in Figure 2-R4. These data are then binned, interpolated and regularized into a number of offset

classes, which are Kirchhoff migrated and finally stacked to create the pre-stack depth migration (PSDM) image. Using the migration velocity, this PSDM image is demigrated to create synthesized shot gathers with ghosts (Figure 2-S2) and without ghosts (Figure 2-S3) using the real source and receiver positions in the training area. A number of shot gather pairs (S2–S3) are used in the training process. We use the terminology *training area* for the area in the PSDM image that is used to create synthesized data, later used as training data. The *prediction area* is where we apply the network. We can use identical training and prediction areas or use a subset of the prediction area as the training area. Moreover, the network can either be used on the data from the same acquisition (step R2 in Figure 2) or another acquisition.

### Kirchhoff demigration

Santos et al. (2000a) show how to use Kirchhoff demigration to compute what we term *synthesized seismograms*. True-amplitude Kirchhoff demigration can be regarded as the inverse process (Hubral et al., 1996; Santos et al., 2000b) or adjoint process (Schuster, 1993) of true-amplitude Kirchhoff migration, where it is described as seismic modelling with the Kirchhoff integral. It involves returning



**FIGURE 3** An illustration showing how to create data with ghosts using Kirchhoff demigration from a PSDM image. The source and receivers are placed above the sea surface (dotted black line) to create the ghosts. Water velocity is used above the sea surface.

from a true-amplitude depth migrated section to the original pre-migration common offset data in time. Demigration has particularly suited for machine-learning-based training because the synthesized data are similar to the original data going into the migration. We use a PSDM-stacked image, a smooth velocity model and Kirchhoff demigration to create synthesized training data. With Kirchhoff demigration, we can create synthesized data for any acquisition geometry. We create the source, receiver and source–receiver ghosts by placing the source and receiver in a mirror position above the sea surface as illustrated in Figure 3. Later, we multiply by the sea surface reflectivity to get the correct polarity and amplitude. A ghosted shot gather,  $D_i(x_j, t_k)$ , is modelled by a linear combination of four ghost-free shot gathers ( $P_i(x_j, t_k)$ ,  $P_i^{SG}(x_j, t_k)$ ,  $P_i^{RG}(x_j, t_k)$ ,  $P_i^{SRG}(x_j, t_k)$ ):

$$D_i(x_j, t_k) = P_i(x_j, t_k) + RP_i^{SG}(x_j, t_k) + RP_i^{RG}(x_j, t_k) + R^2 P_i^{SRG}(x_j, t_k), \quad (2)$$

where  $P$  is the dataset using the actual source and receiver locations,  $P^{SG}$  is the dataset using the mirror source location,  $P^{RG}$  is the dataset using the mirror receiver location,  $P^{SRG}$  is the dataset using the source and receiver mirror locations,  $R$  is the reflectivity of the water surface,  $i$  is the shot number,  $x_j$  is the offset and  $t_k$  is the time. This means that we create four datasets: (1) the ghost-free primary,  $P$ , by using the original source and receiver positions (black ray in Figure 3), (2) the source ghost,  $P^{SG}$ , by placing a source at the mirror position above the sea surface (red ray in Figure 3), (3) the receiver ghost,  $P^{RG}$ , by placing a receiver at the mirror position above the sea surface (green ray in Figure 3) and (4) the source–receiver ghost,  $P^{SRG}$  by placing both a receiver and a source at the mirror positions above the sea surface (white ray in Figure 3). The reflectivity of the water surface,  $R$ , results in a polarity shift when multiplied with arrivals (2)

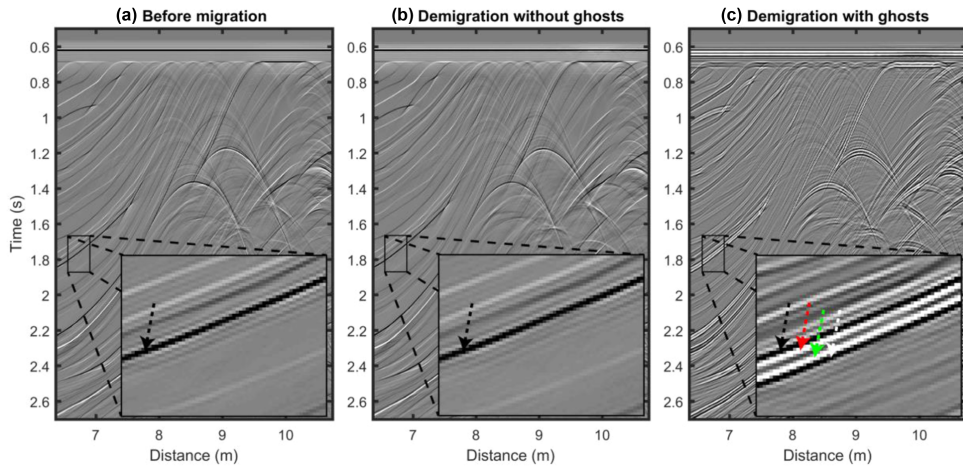
and (3). The ghost-free dataset is  $P$ , while the dataset with ghosts is created using Equation (2), which is a linear combination of all four datasets. As a result, we get synthesized data with and without ghosts as shown in Figure 4b,c. During the training, the input to the network is synthesized shot gathers with ghosts (Figure 5a), and the outputs are synthesized shot gathers without ghosts (Figure 5b).

## Convolutional neural networks

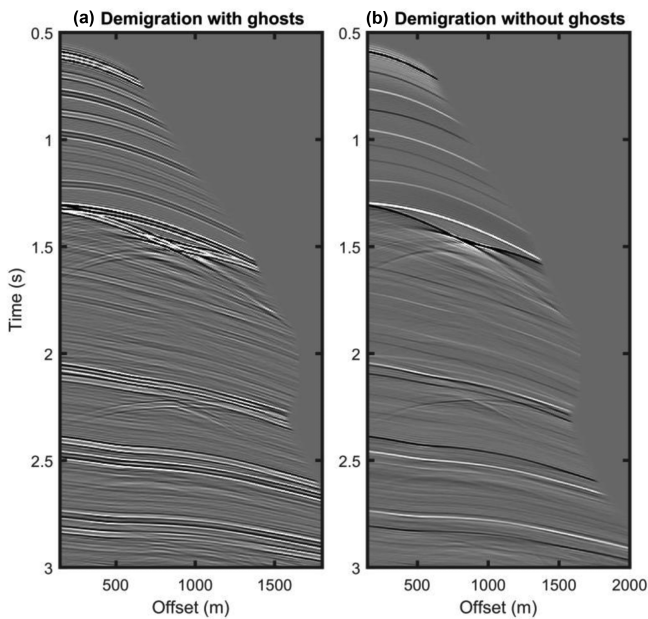
A convolutional neural network (CNN) uses convolutions instead of general matrix multiplications (Goodfellow et al., 2016). By doing this, the network has sparse interactions between the neurons in the network. In particular, the kernel size determines the number of interactions from one layer to the next. A CNN is typically used when it is assumed that the meaningful features are local. As a result, sparse interactions can save memory and computational cost.

In this paper, we used a CNN structure called a U-net (Ronneberger et al., 2015). This structure is made up of an encoder and decoder with skip connections. The encoder downsamples the number of pixels, and the decoder upsamples the number of pixels. Various pooling functions (maximum, average, L2, etc.) can be used to downsample the number of pixels. In contrast, a transposed convolution can be used to upsample the number of pixels. Transposed convolution is an operation that goes in the opposite direction of a normal convolution. Dumoulin and Visin (2016) show several examples of transposed convolution. The U-net is different from an encoder–decoder because it contains skip connections that copy feature maps from one layer to another. Since the U-net uses downsampling and consequently reduces the image size, it is also more efficient than a CNN with no downsampling. Another benefit of the U-net downsampling is the increased receptive field of the network (Lucas et al., 2018). In contrast, only the size of the convolutional kernel determines the receptive field of a CNN with no downsampling.

The structure of our U-net has also been used in de Jonge et al. (2021) showing good results. Furthermore, the structure is shown in Figure 6 and the hyperparameters are shown in Table 1. In addition, we compare this U-net with the Dunet presented by Peng et al. (2021). The Dunet has many structural similarities with the U-net. More information about the Dunet can be found in Peng et al. (2021). These two networks are used for all our tests. That said, the goal of this paper is not to find the optimal network structure, rather it is focused on creating training data that can be used to get high-quality deghosted data. We believe that training data are one of the most important aspects when using a neural network to achieve good deghosting results.



**FIGURE 4** Common channel gathers before migration (a), demigration without ghosts (b), and demigration with ghosts (c). Data before migration are modelled with FD. A zoom is used in all three plots focusing on a single event where arrows highlight the primary (black), source ghost (red), receiver ghost (green) and source–receiver ghost (white).

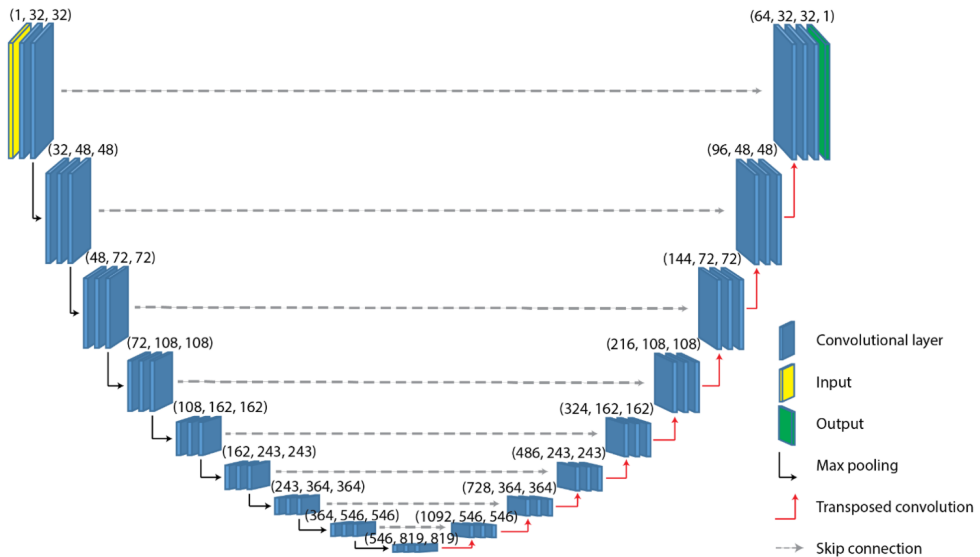


**FIGURE 5** Demigrated shot gather with ghosts (a) and without ghosts (b).

## RESULTS

This section is divided into two parts: (1) synthetic data, which focuses on various tests to understand the problems and

advantages of DEGDEM (DEGhosting using DEMigration-based supervised learning), and (2) real data, which focuses on one specific example from the Tampen area in the North Sea.



**FIGURE 6** The U-net structure used in this paper. The numbers represent the number of feature maps at each stage in the network. The hyperparameters are shown in Table 1.

**TABLE 1** Hyperparameters used in the U-net shown in Figure 4

Loss function	Mean square error
Optimization	Adam (learning rate = 0.0001)
Kernel sizes for convolution	3×3
Kernel size for max pooling	2×2
Kernel size for transposed convolution	2×2
Stride for convolution	1×1
Stride for max pooling	2×2
Stride for transposed convolution	2×2
Activation function	ReLU
Batch size	4
Epochs	200

## Synthetic data

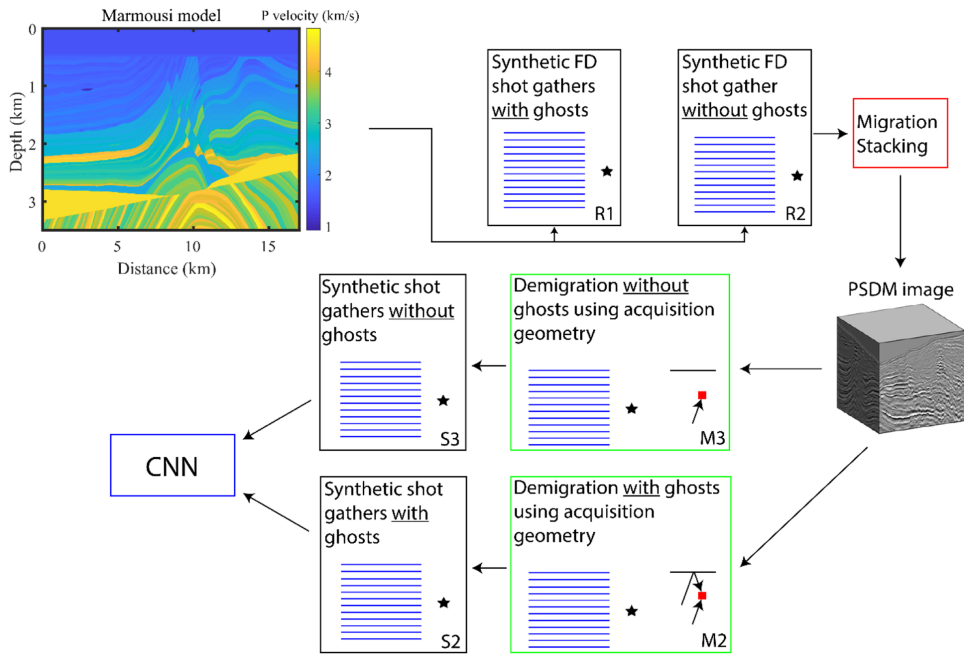
Evaluating the quality of the deghosting results is often difficult on real data because we do not know the true ghost-free data. For that reason, we have conducted a synthetic analysis. In addition, this synthetic analysis can tell us more about the advantages and limitations of using DEGDEM. In the synthetic analysis, we used the Marmousi model (Martin et al., 2006) and acoustic finite-difference (FD) modelling to model data with and without the ghosts. The DEGDEM workflow for the synthetic analysis is shown in Figure 7 and is similar to the workflow shown in Figure 2. The FD data were modelled using the Marmousi P-wave velocity and density

(Martin et al., 2006) shown in Figure 8. The source is placed at 6 m depth and a streamer with hydrophones at 20 m depth. The receiver and shot increments are 12.5 and 6.25 m, respectively. The offset to the first receiver is 147 m. Mirror locations above the sea surface for the source and receiver were used to model the ghosts as described in the section ‘Methodology – Kirchhoff demigration’. As a result, we have two datasets – one with ghosts and one without.

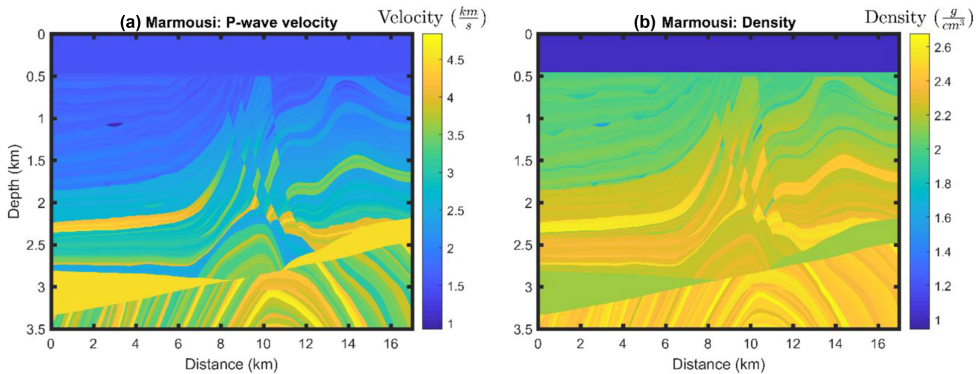
A smooth velocity model (Figure 9a) was used for the travel time calculations in Kirchhoff demigration. This velocity model was a smooth version of the Marmousi model shown in Figure 8. We Kirchhoff migrated and stacked the FD modelled data without ghosts to generate a pre-stack depth migration (PSDM) image shown in Figure 9b. Both the smooth velocity model and the PSDM image were used to create synthesized data with Kirchhoff demigration. The mirror source and mirror receiver technique described in the ‘Methodology – Kirchhoff demigration’ subsection was used to create data with and without ghosts that were used for training the neural network.

In this subsection, we describe several tests to identify some advantages and disadvantages of DEGDEM. The first test, entitled baseline test (first row in Table 2), relates to a neural network trained and tested on ghosts that were modelled using the correct source locations, receiver locations and sea surface reflection coefficient. The second and third tests, entitled the swell wave test and the reflection coefficient test (second and





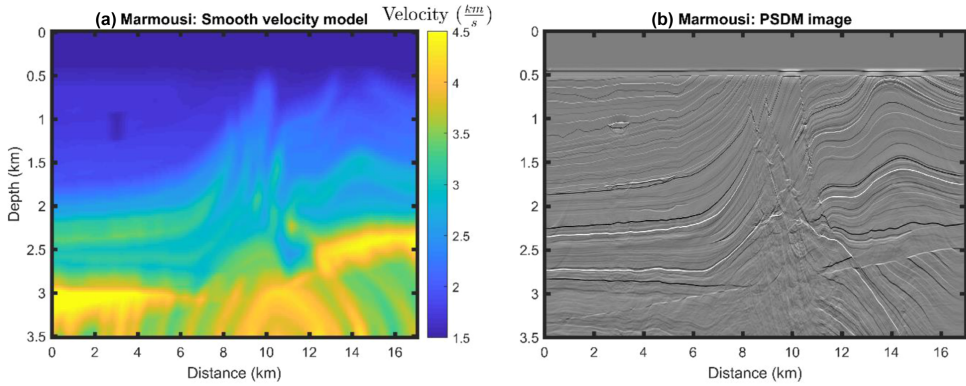
**FIGURE 7** An illustration of the DEGDEM workflow used for the synthetic tests. First synthetic data with and without ghosts are modelled using FD and the Marmousi model. The ghost-free data are migrated to obtain the PSDM image. The PSDM image together with Kirchhoff demigration is used to generate synthesized shot gathers with and without ghosts.



**FIGURE 8** The Marmousi models for P-wave velocity (a) and density (b) are used to generate synthetic data.

third rows in Table 2), examine the sensitivity of the network used in the first test to swell waves and changes in the sea surface reflection coefficient. In both the second and third tests, we also re-train the U-net and DUnet using different training data attempting to make the network more robust to swell waves and changes in sea surface reflectivity. The fourth

test, entitled residual ghost (fourth row in Table 2), uses a PSDM image containing residual ghost to generate training data for DEGDEM. This test investigates how sensitive the network is to residual ghost in the PSDM image. The fifth test, entitled multiple test (fifth row in Table 2), investigates if the network can attenuate ghosts of surface-related



**FIGURE 9** Smooth P-wave velocity model (a) and PSDM image (b). Both of these models are used to create synthesized data with Kirchhoff demigration.

**TABLE 2** NRMS error (common channel domain, channel 15) when using conventional deghosting, standard DEGDEM (U-net or DUnet) for deghosting, or generalized DEGDEM (U-net or DUnet) for deghosting. Each result is separated into the various tests in this subsection

	Conventional	Standard DEGDEM (U-net)	Standard DEGDEM (DUnet)	Generalized DEGDEM (U-net)	Generalized DEGDEM (DUnet)
Baseline test	0.0030	0.0032	0.0033	N.A.	N.A.
Swell wave test	0.0171	0.0044	0.0064	0.0039	0.0078
Reflection coefficient test	0.0058	0.0039	0.0050	0.0031	0.0028
Residual ghost	0.0171	0.0070	0.0081	N.A.	N.A.
Multiple test	0.0040	0.0042	0.0040	N.A.	N.A.

multiples. This test is important because the synthesized data do not contain surface multiples, in contrast to real data. A summary of all results from this subsection is shown in Figure 10 and Table 2, while more detailed descriptions of the tests are given in their respective subsections. Here, a generalized DEGDEM represents a network that has been trained to be more robust to swell waves or changes in sea surface reflection coefficient.

To quality control our results, we use common channel gathers (channel 15) before deghosting, after deghosting, and the difference. We evaluate a normalized root mean square (NRMS) error measurement in the common channel domain given by the equation:

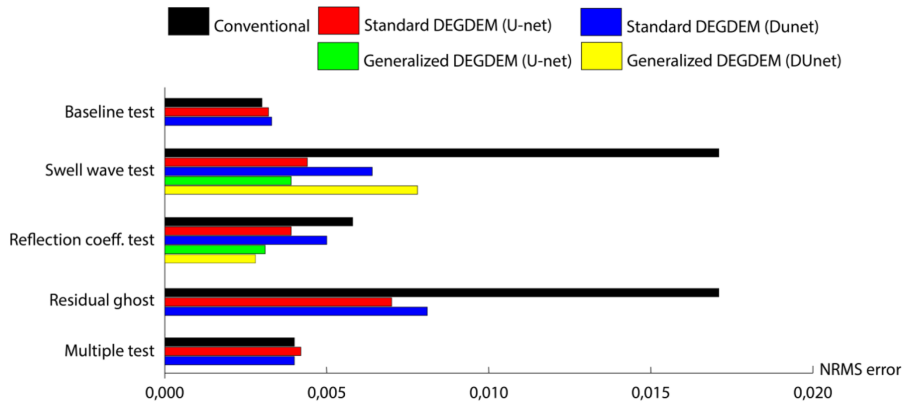
$$e^{\text{NRMS}} = \frac{e^{\text{RMS}}}{y_{\text{max}} - y_{\text{min}}} = \frac{1}{y_{\text{max}} - y_{\text{min}}} \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}, \quad (3)$$

where  $y_i$  is the data without ghosts,  $\hat{y}_i$  is the deghosted data and  $n$  is the number of samples.

### Baseline test

In this test, we trained a network using the DEGDEM workflow shown in Figure 7 using the correct source locations, receiver locations and sea surface reflection coefficient. We used 2340 pairs of synthesized training shot gathers (with and without ghosts) to train the CNN. Afterwards, we applied the CNN on 2340 of FD shot gathers (with ghosts) within the training area and 321 FD shot gathers outside the training area. In the next subsections, all synthetic tests use the same training area and prediction area as in the baseline test. We ran 200 epochs in this test and ensuing tests. We compared the neural network result with the conventional deghosting method of Poole (2013). Figure 11 shows the results using conventional deghosting and DEGDEM (U-net) deghosting within the training area. DEGDEM (U-net), DEGDEM (DUnet) and conventional deghosting NRMS errors were 0.0032, 0.0033 and 0.0030, respectively. The NRMS error was calculated using all samples from 0.5–3 s and 2002–16,607 m. All synthetic tests in the following subsections use the same window





**FIGURE 10** Bar graph showing NRMS error (common channel domain, channel 15) when using conventional deghosting, a standard DEGDEM (U-net or DUnet) for deghosting or a generalized DEGDEM (U-net or DUnet) for deghosting. Each result is separated into the various tests in this subsection.

to calculate the NRMS error. The amplitude spectrum in this test and all ensuing synthetic tests were calculated using the same window as the NRMS error. Figure 11 and the NRMS errors demonstrate that DEGDEM and conventional deghosting both have low error. Referring to the amplitude spectra, in general the conventional deghosting worked better than DEGDEM at the ghost peak frequencies, and DEGDEM performed better at the ghost notches (except for the last notch at approximately 75 Hz). DEGDEM may have performed better than conventional deghosting in the ghost notches because the training data were similar to the prediction data, which is one of the advantages when using demigration to create training data. In other words, it was already familiar with the ‘spiky’ nature of the prediction input. The conventional deghosting, on the other hand, was more general and did not have a bias towards any wavelet or geology. We also observe that DEGDEM performed less well in the area of conflicting dips in the centre of the section compared to the DEGDEM in areas of less complex geology. In this perfect scenario, the conventional method performed better than DEGDEM in general. DEGDEM (U-net) was used on FD data outside the training area to demonstrate that the network works on data with similar geology. The results are shown in Figure 12, where the training area is to the right of the dashed line. The NRMS error is shown for each shot and does not change significantly when DEGDEM is applied to data outside the training area. As an additional test, we exclude the central part of the Marmousi model as training data (shown in Figure 13). Data outside the dashed lines were used as training data. The central part of the model contained more complex geology with steeply dipping layers and faults. The amount of training data was still 2340 shot gather pairs by including the test data from the previous test (shown in Figure 12). The results are shown in Figure 13,

where the NRMS error does not change significantly in the central part of the model.

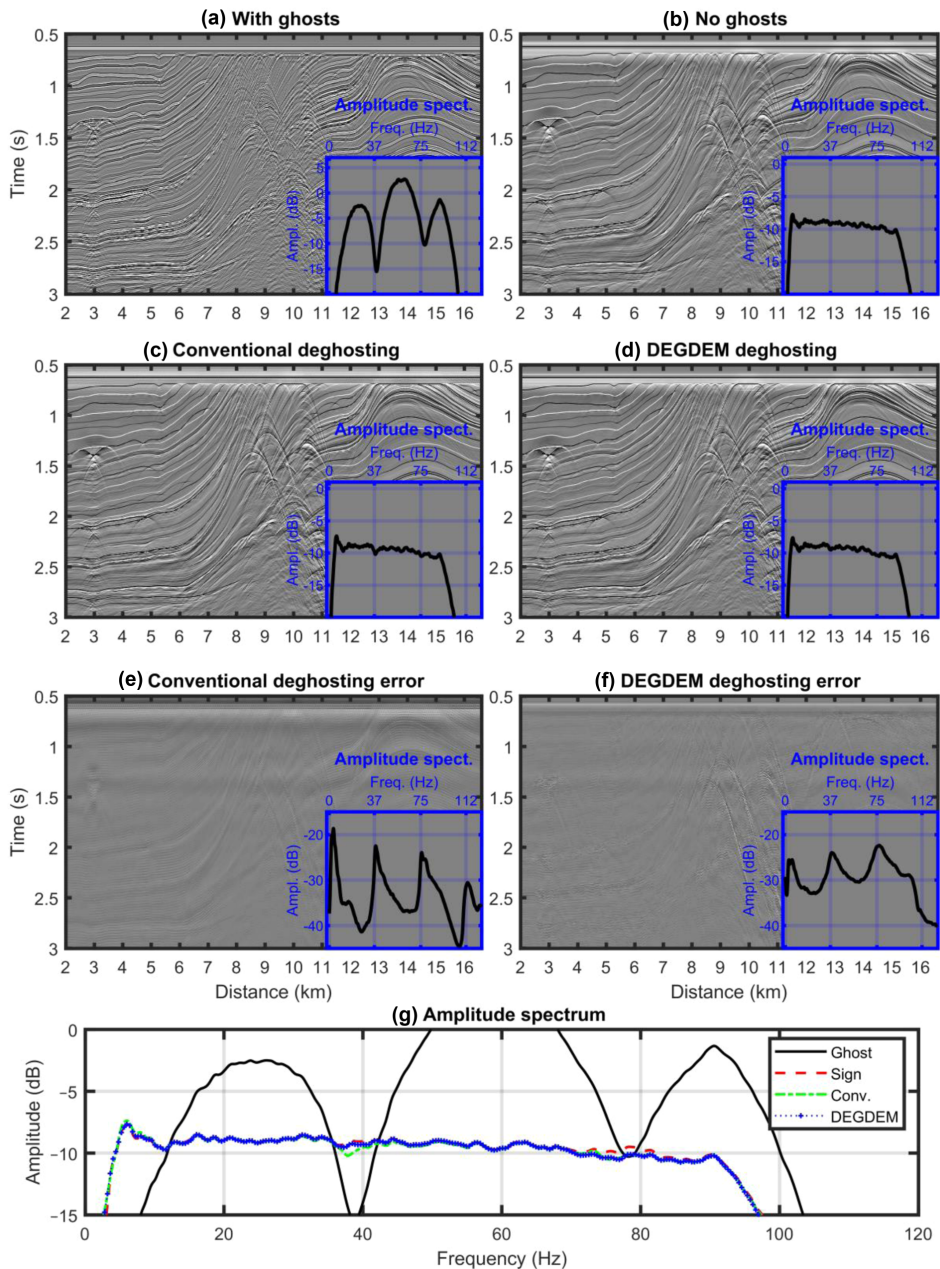
### Sensitivity and generalization of streamer depth and sea surface reflection coefficient

Ocean waves, bad weather and the imperfect positioning of receivers are a natural part of a marine seismic acquisition. The sea surface reflection coefficient may also change from shot to shot or with offset and is not always known. Consequently, a ghost model based on nominal reflectivity coefficients and receiver depths may be inaccurate. We will investigate how robust the neural network is to these changes. A key feature of neural networks is their adaptability. Consequently, it may be possible to train a neural network on many different streamer depths and sea surface reflection coefficients. By using this approach, it could be possible to generalize the network. A generalized network would be a big advantage and easy to use on real data. We use the network from the ‘Baseline test’ subsection and train new networks that have been generalized on either streamer depth or sea surface reflection coefficient.

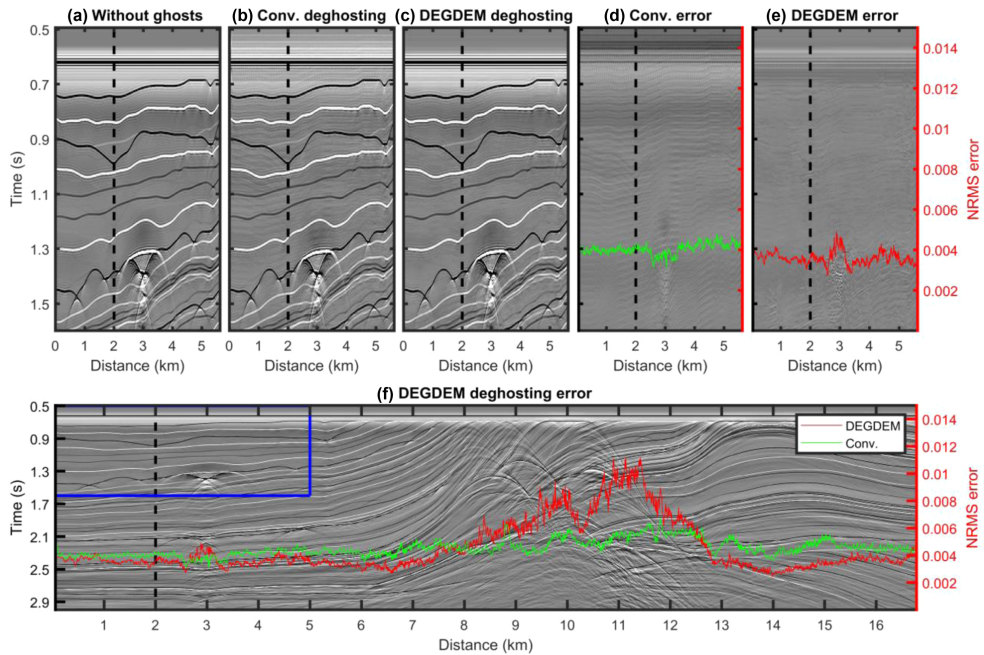
We start by focusing on streamer depth. In our test, we simulated a swell wave that changes the height of the water column above the receivers as shown in Figure 14. For simplification, we model the swell wave as a sinusoid wave:

$$A(x) = B \sin(k(x - \phi)), \quad (4)$$

where  $B$  is the amplitude,  $k$  is the wavenumber,  $x$  is the offset and  $\phi$  is the phase shift. The swell wave parameters are



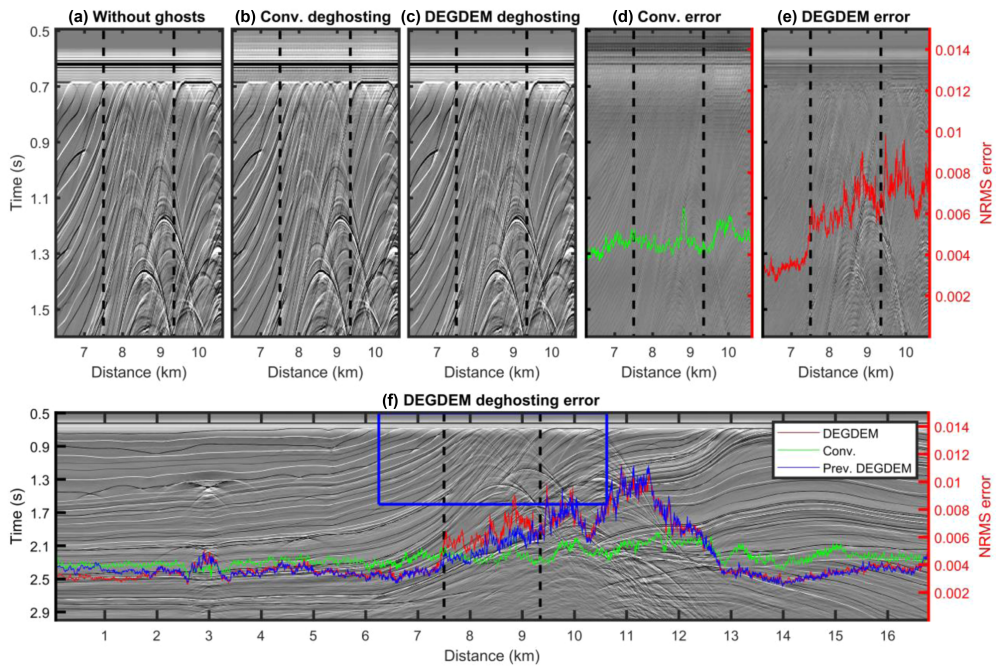
**FIGURE 11** Common channel gather (channel 15) displays for (a) data with ghosts; (b) data without ghosts; (c) conventional deghosting; (d) DEGDEM deghosting; (e) residual noise left after conventional deghosting (difference between no ghosts and conventional deghosting); (f) residual noise left after DEGDEM deghosting; (g) amplitude spectrum of data with ghosts (black solid line), without ghosts (red dashed line), conventional deghosting (green dashed line) and DEGDEM deghosting (blue dashed line).



**FIGURE 12** Common channel gathers (receiver 15) before deghosting (a), conventional deghosting (g), DEGDEM deghosting (c), conventional deghosting error (d), DEGDEM deghosting error (e) and a zoom of the DEGDEM deghosting (f). The area shown in subfigures a, b, c, d and e is indicated by a blue rectangle in subfigure f. Right of the dashed line is used as a training area. The red and green lines show the NRMS error using DEGDEM and conventional deghosting, respectively.

shown in Figure 14. These parameters are chosen to be realistic to swell waves during acquisition. Ideally, we would do FD modelling with a ‘wavy’ curved upper surface as shown in Figure 14. However, it is difficult, in practice, to do FD modelling with a curved surface to create a ghost model that changes as a function of offset. Instead, we approximate the variability in the ghost model by changing the vertical receiver ghost positions as shown in Figure 14. We follow a similar workflow to create synthesized data from the section ‘Methodology – Kirchhoff demigration’. However, the vertical positions of receivers above the still-water line change as a function of offset and also change for each shot, as shown in Figures 14 and 15. The swell wave changes for each shot by using a phase shift from Equation (3) that depends on the shot increment and relative wave speed. We assume that the source ghost does not change during this test and acquisition in general. The air-gun array is often positioned close to the sea surface and follows the vertical motion of the waves. This scenario generates a receiver ghost that changes with offset and from shot to shot, as shown in Figure 15c,d. The swell wave moves slowly compared to the seismic waves so we can assume that the sea surface does not change within

a shot record. We assume furthermore that the tilt of the sea surface is negligible because of the small ratio between the wave height and wavelength. Modelling a ghost this way is a simplification of a real marine environment. A more realistic dynamic sea surface could be modelled using the approach by Blacquièrre and Sertlek (2019). However, this test should be realistic enough to help understand how sensitive the network is to changes in the ghost model. The effect of these swell waves on the receiver ghost is visible on a shot gather and a common channel gather shown in Figure 15. We first test the network trained on a constant streamer depth of 20 m (from the ‘Baseline test’ subsection) and the conventional deghosting method by Poole (2013) using parameters consistent with a constant streamer depth of 20 m. The NRMS error using the DEGDEM (U-net) deghosting, DEGDEM (DU-net) deghosting and conventional deghosting is 0.0044, 0.0064, and 0.0171, respectively. Figure 16 shows the DEGDEM (U-net) and conventional deghosting results and errors using both methods. Both DEGDEM and the conventional deghosting leave residual ghost that appears as a ‘grainy’ pattern in Figure 16, because of the cyclical period of the swells. To make DEGDEM more robust to errors in streamer



**FIGURE 13** Common channel gathers (receiver 15) before deghosting (a), conventional deghosting (b), DEGDDEM deghosting (c), the conventional deghosting error (d), the DEGDDEM deghosting error (e) and a zoom out of the DEGDDEM deghosting (f). The area shown in subfigures a, b, c, d and e is indicated by a blue rectangle in subfigure f. The area between the dashed lines is not used as training data. The red, green and blue lines show the NRMS error using DEGDDEM, conventional deghosting and the previous DEGDDEM network (shown in Figure 12), respectively.

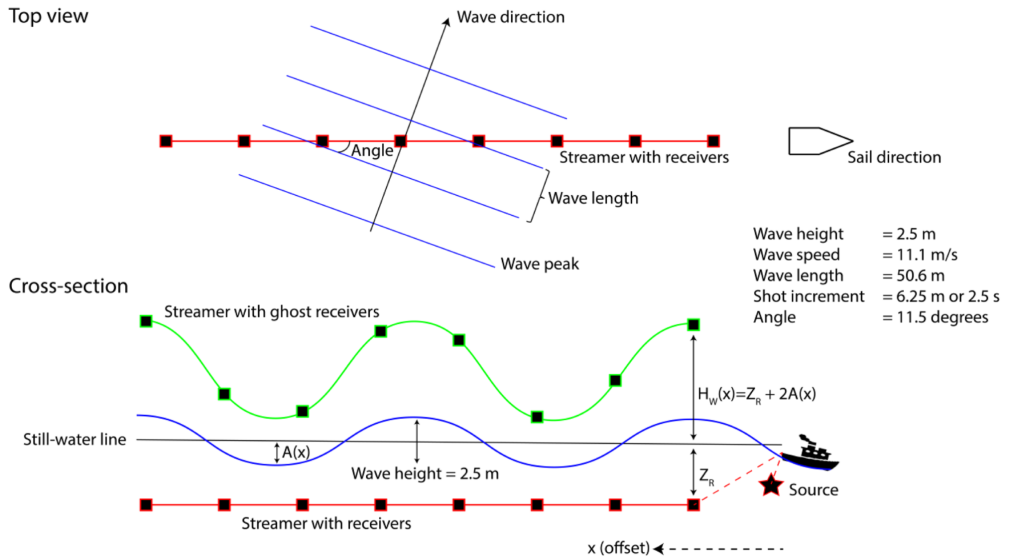
depth, we trained a U-net and a DUnet on flat streamers with depths ranging from 18 to 22 m with a 0.5-m increment. The NRMS error using the U-net and DUnet are 0.0039 and 0.0078, respectively. The U-net shows an improvement from training only on one streamer depth. However, the DUnet showed a worse result, which indicated that this structure might not be ideal when making a robust network to swell waves. There are endless opportunities to create training data with streamer depth perturbations. In this example, we show the simplest perturbation. However, a constant streamer depth perturbation is sufficient to make the U-net more robust to changes in streamer depth due to swells. The results also show that both networks are less sensitive to changes in streamer depth compared to conventional deghosting. Note that it is possible to incorporate a variable sea surface datum in conventional deghosting (see, e.g., King & Poole, 2015). However, this could complicate the workflow as it requires the computation of wave heights prior to the deghosting.

Second, a simple test was done to test the sensitivity and generalization of the sea surface reflection coefficient. The real reflection coefficient will change with incidence angle

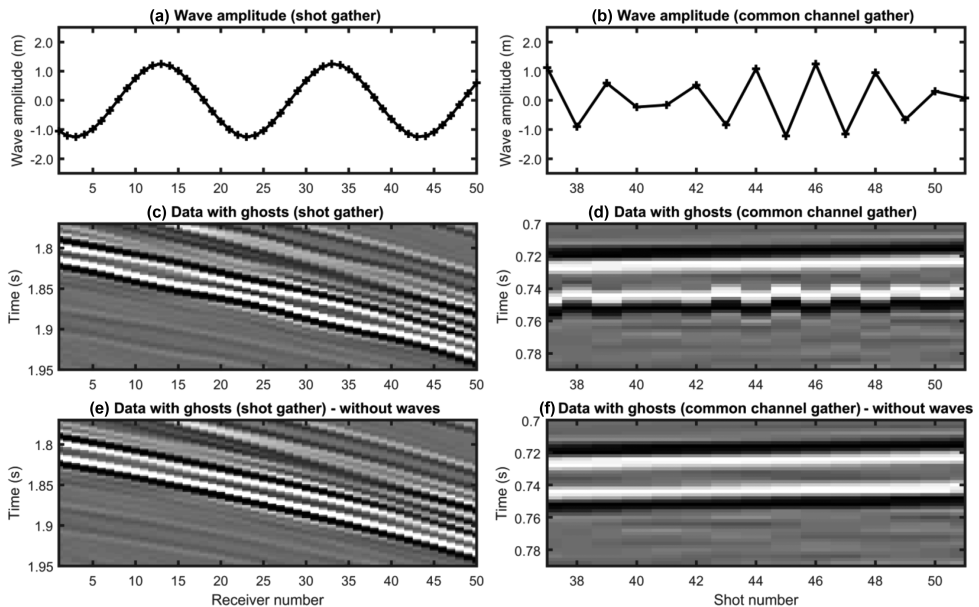
and frequency and is dependent on the sea state (Orji et al., 2013; Asgedom et al., 2017), which makes it complicated to estimate. We simplify by assuming that the reflection coefficient is constant. We generate FD data with a reflection coefficient of  $-0.92$  for both the source and receiver ghosts. We test the baseline networks that are trained on data with a reflection coefficient of  $-1$  from the subsection ‘Baseline test’. We compare these networks with the conventional deghosting method by Poole (2013) using parameters consistent with a reflection coefficient of  $-1$ . The NRMS error using DEGDDEM (U-net), DEGDDEM (DUnet) and conventional deghosting is 0.0039, 0.0050, and 0.0058, respectively. Moreover, a generalized U-net and DUnet were trained on reflection coefficients ranging from  $-0.90$  to  $-1$  with an increment of 0.02. The NRMS error using the U-net and DUnet was 0.0031 and 0.0028, respectively. These results indicate that the networks were less sensitive than the conventional method to the change in sea surface reflection coefficient. In addition, the result also showed a significant improvement if the networks were generalized.

Water velocity can also change during the acquisition and deviate from the estimated water velocity, consequently





**FIGURE 14** Illustration showing how receiver depth changes with offset and between shots because of swell waves. It also demonstrates how we create the receiver ghost by changing the ghost receiver height ( $H_W$ ) above the still-water line.



**FIGURE 15** (a) The swell wave amplitude above the still-water line as a function of receiver number. (b) The swell wave amplitude above the still-water line as a function of shot number. (c) Shot gather with source and receiver ghosts. (d) Common channel gather with source and receiver ghosts. (e) and (f) show how the data would look like without the swell wave in a shot gather and a common channel gather, respectively. The top black and white events are the primary and source ghost, respectively. The following white and black events are the receiver and source-receiver ghosts, respectively.



changing the time lag of the ghosts. However, it is likely that a generalized DEGDEM would be able to account for these changes equally well as changes in timing caused by other factors (such as swells).

### Sensitivity to residual ghost in the reflectivity model

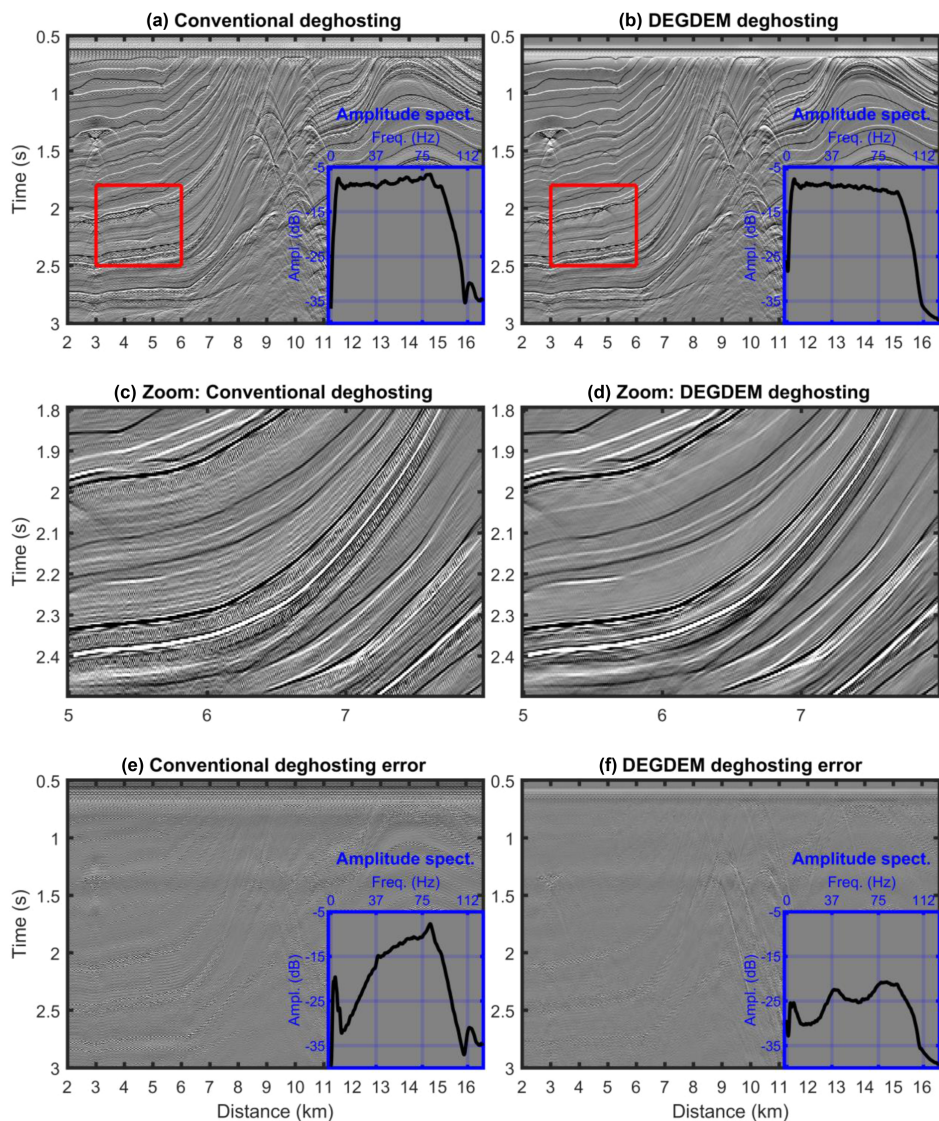
One potential limitation of the workflow shown in Figure 2 is the presence of a residual ghost in the data before migration. The residual ghost is caused by imperfect deghosting and will be included in the PSDM image after migration and, consequently, the synthesized demigrated training data. To investigate this, we apply a deliberately imperfect deghosting on the synthetic data, which will lead to residual ghosts in the PSDM image. The first part of the workflow shown in Figure 7 is changed to the workflow shown in Figure 17.

We use synthetic data with swell waves from the previous subsection ‘Sensitivity and generalization to streamer depth and sea surface reflection coefficient’. We deghosted the data using conventional deghosting assuming 20 m streamer depth, which led to imperfect deghosting and residual ghosts as shown in Figure 16c. This deghosted data ended up in the PSDM image, and we created synthesized data using Kirchhoff demigration with a 20-m streamer depth. These data were used to train the U-net and DUNet that were applied to data with swell waves. The result using the DEGDEM (U-net) deghosting and DEGDEM (DUNet) deghosting gave an NRMS error of 0.0070 and 0.0081, respectively. The error using conventional deghosting and DEGDEM (U-net) are shown in Figure 18. As shown in the previous subsection, the NRMS error using conventional deghosting and a U-net trained on data from a clean reflectivity model (but 20 m constant streamer depth) were 0.0171 and 0.0044 (Figure 16), respectively. These results show an improvement from the conventional method. However, these results also show that the networks are sensitive to residual ghost in the PSDM image.

### Surface-related multiples in the test data

Up to this point, we have shown examples of primary reflections (which correspond to only one reflection from each reflector or diffractor in the subsurface) containing ghosts. However, real data also contains multiple reflections (which experience more than one reflection from the subsurface) which also have ghosts. One potential limitation of the

DEGDEM workflow (Figure 2) is that Kirchhoff demigration does not consider all waves which have ghosts, such as converted waves or multiples. Surface and internal multiples are problematic because they interfere with primary reflections. As a result, suppression of multiples is an essential prerequisite for accurate seismic imaging and interpretation. We will test how the network reacts when exposed to multiples in the synthetic FD data. One option to model all surface-related multiples is to use a free surface boundary condition in the FD modelling. However, with this option, we cannot generate data with multiples and without ghosts, which we need for our ground truth. In addition, if we use a free surface boundary condition, we cannot remove the multiples after deghosting to identify any primary damage caused by the deghosting. Therefore, we avoid a free surface boundary condition and generate the first-order surface-related multiples by first placing the source 900 m (twice the water depth of 450 m) above its original position and create synthetic FD data. Similarly, we place the receivers 900 m above their original position and create synthetic FD data. These two datasets are combined and multiplied by the sea surface reflection coefficient ( $-1$ ) and the zero-offset reflection coefficient at the water bottom to create a close approximation to first-order multiples. Most of the multiple energy is caused by the first-order multiple. The difference is small between our multiple models and the multiples we get using a free surface boundary condition in the FD modelling. The data with and without multiples included are shown in Figure 19a,b with blue arrows highlighting surface-related multiples. We use the same networks as in the subsection ‘Baseline test’ and apply them to this data with surface-related multiples. Figure 19 shows the results using DEGDEM (U-net) and conventional deghosting. After deghosting, we remove the surface-related multiples to investigate if there is any primary damage or imperfect deghosting caused by the multiples (Figure 19e,f). Visually, we observe the source, receiver and source–receiver ghosts removed from the data using either DEGDEM or conventional deghosting. In addition, the primaries seem to be preserved at locations where primary and multiple overlaps. However, the blue arrow at 2 s in Figure 19g,h shows some error in deghosting. The NRMS error using DEGDEM (U-net), DEGDEM (DUNet) and conventional deghosting before demultiple is 0.0042, 0.0040 and 0.0040, respectively. As in the baseline test, the conventional deghost worked better than DEGDEM at the ghost peak frequencies, and DEGDEM performed better at the ghost notches. This simple test demonstrates that the networks can remove ghosts of multiples. A combination of source- and receiver-side multiples along with other multiple generators would complicate this analysis.



**FIGURE 16** (a) Conventional deghosting, (b) DEGDEM deghosting, (c) Conventional deghosting zoom (shown as a red rectangle in subfigure a), (d) DEGDEM deghosting zoom, (e) conventional deghosting error (NRMS = 0.0171) and (f) DEGDEM deghosting error (NRMS = 0.0044).

### Real data: PL988 Tampen

The real data used in this paper are from a survey located in the North Sea, Tampen area (PL988) off the western coast of Norway. CGG acquired these data in 2015 and 2016. The survey used a variable streamer depth configuration

(Soubaras & Dowle, 2010; Soubaras et al., 2012) with dual-level source arrays (Siliqi et al., 2013) as shown in Figure 20. In total, the survey has 12 streamers and two sources. We have data before deghosting, a PSDM image and a smooth velocity model. Using the survey geometry for all cables and both sources, we generated a trained CNN model using DEGDEM

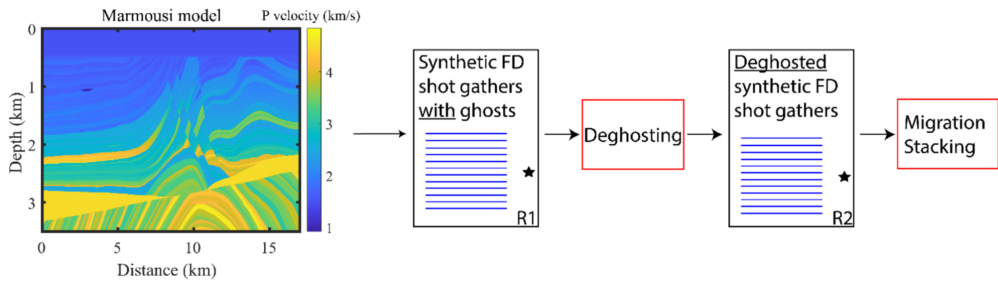


FIGURE 17 Workflow used to test the sensitivity to residual ghost in the PSDM image.

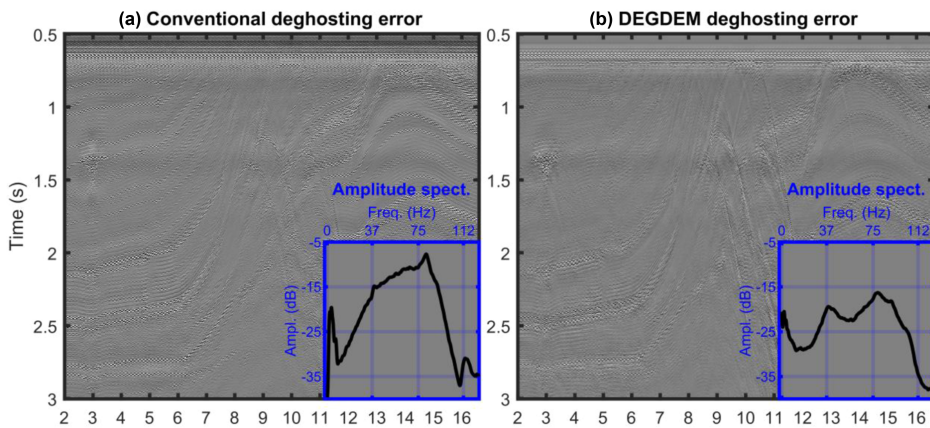


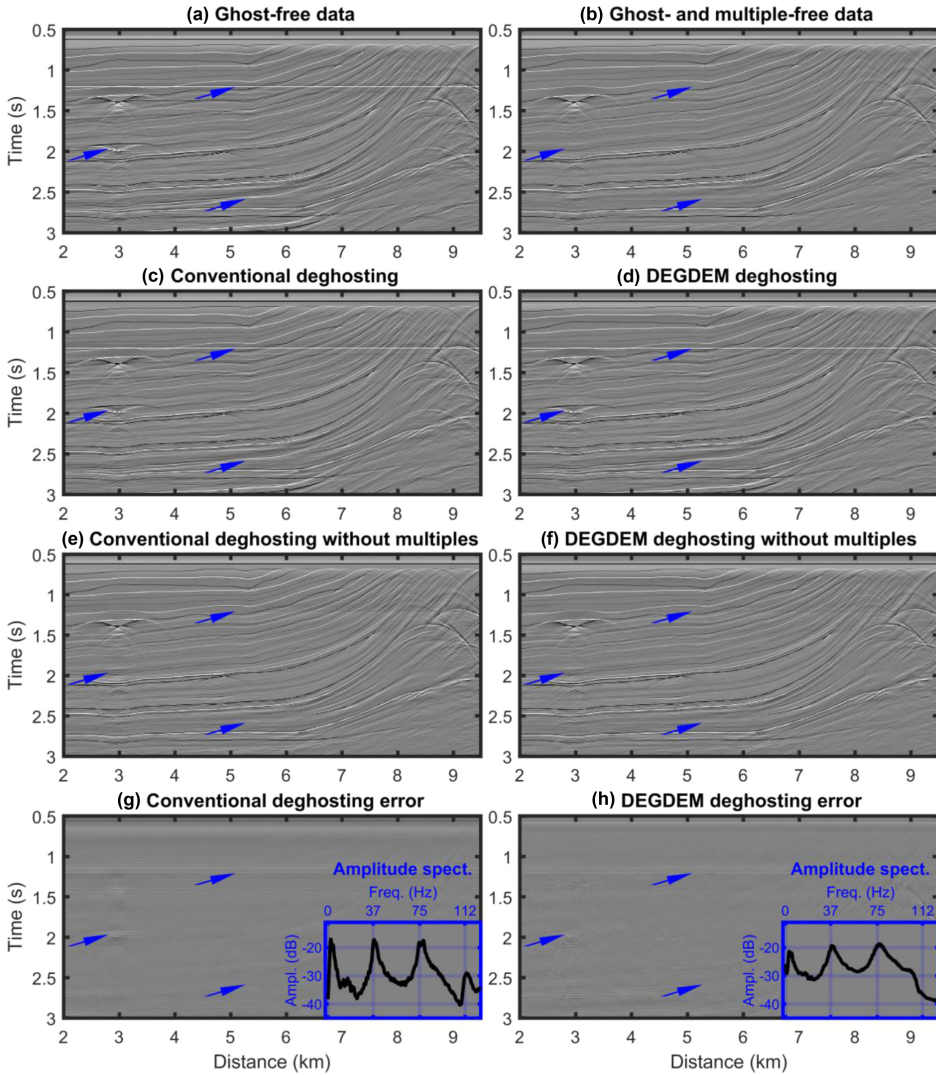
FIGURE 18 (a) Conventional deghosting error (NRMS = 0.0171) and (b) DEGDEM deghosting error (NRMS = 0.0070).

and applied the network on the real data before deghosting and compared it with conventional deghosting (Poole, 2013). We assumed a single level source at 6 m for conventional deghosting and DEGDEM for simplicity purposes. Due to this assumption, both deghosting results will be suboptimal. The Discussion section will highlight that the resolution of some of these features is significantly improved by using conventional deghosting with a dual-level source.

We show our results in common channel gathers. This domain was chosen because we have variable streamer depth data and in this domain, we can clearly observe the ghost notches in the amplitude spectrum. In addition, the time delay from the primary to the ghosts is approximately the same. We show our results in channel 15 and channel 50 for source 1 and streamer 6 (Figure 20). The offset, depth and notch frequency for channel 15 are approximately 324 m, 10.5 m and 71 Hz, respectively. For channel 50, the offset, depth and notch frequency are approximately 762 m, 18.3 m and 41 Hz, respectively. We show only results using the U-net.

The DUnet was also used but showed similar results to the U-net and is consequently not shown. The results from channel 15 are shown in Figure 21. The amplitude spectrum in Figure 21g (calculated from 0.3 to 1.5 s) shows that DEGDEM deghosting has more energy around the ghost notch. The results (Figure 21c,d,e,f) show an increased resolution using DEGDEM deghosting. In Figure 21d,f, we also see an event (highlighted by the top-left blue arrow) that has almost disappeared using conventional deghosting but remains visible using DEGDEM deghosting. The blue arrows, in Figure 21d,f, show examples with better-defined reflectors using DEGDEM deghosting. The results from channel 50 are shown in Figure 22. The amplitude spectrum in Figure 22g (calculated from 0.92 to 1.24 s) shows that the DEGDEM deghosting has more energy around the ghost notches. A similar observation was made during the inspection of the synthetic data. Analysis of potential residual ghost energy is challenging in shallow water areas where multiple reverberations cross-cut other arrivals. It will certainly be the

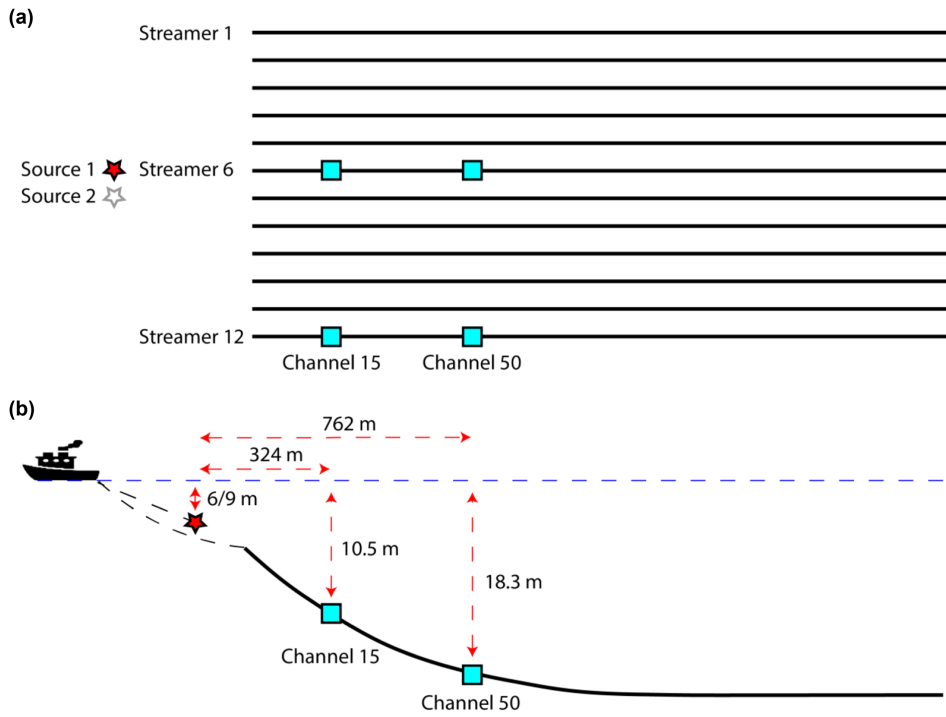




**FIGURE 19** Common channel gathers without ghost and with multiples (a), without ghosts and without multiples (b), conventional deghosting (c), DEGDEM deghosting (d), conventional deghosting with multiples removed (e), DEGDEM deghosting with multiples removed (f), conventional deghosting error (g), and DEGDEM deghosting error (h). Deghosting is done before removing the multiples.

case that in some areas ghost arrivals will coincide with primary or multiple arrivals from other reflectors. The green arrows, in Figure 22a,b, highlight two arrivals with similar timing to anticipated receiver ghost energy. Figure 22c,d,e,f shows some arrivals with similar timing after deghosting. This energy is slightly weaker on the DEGDEM deghosting compared to the conventional deghosting. Figure 23 shows the

results of a shot gather. This domain illustrates that DEGDEM has removed some linear noise and weak diffraction energy. As a result, DEGDEM deghosting appears less noisy. However, given the data used in the DEGDEM training was after migration and demigration, which is known to contain inherently clean data, this result is not surprising. This linear noise is dipping in the opposite direction of the primary and is quite



**FIGURE 20** (a) Top view of the survey geometry and (b) cross section of streamer 6. Channel 15 and channel 50 are highlighted since we look at results from these channel gathers.

different from the training data. As a result, the network might not know how to handle or deghost this noise.

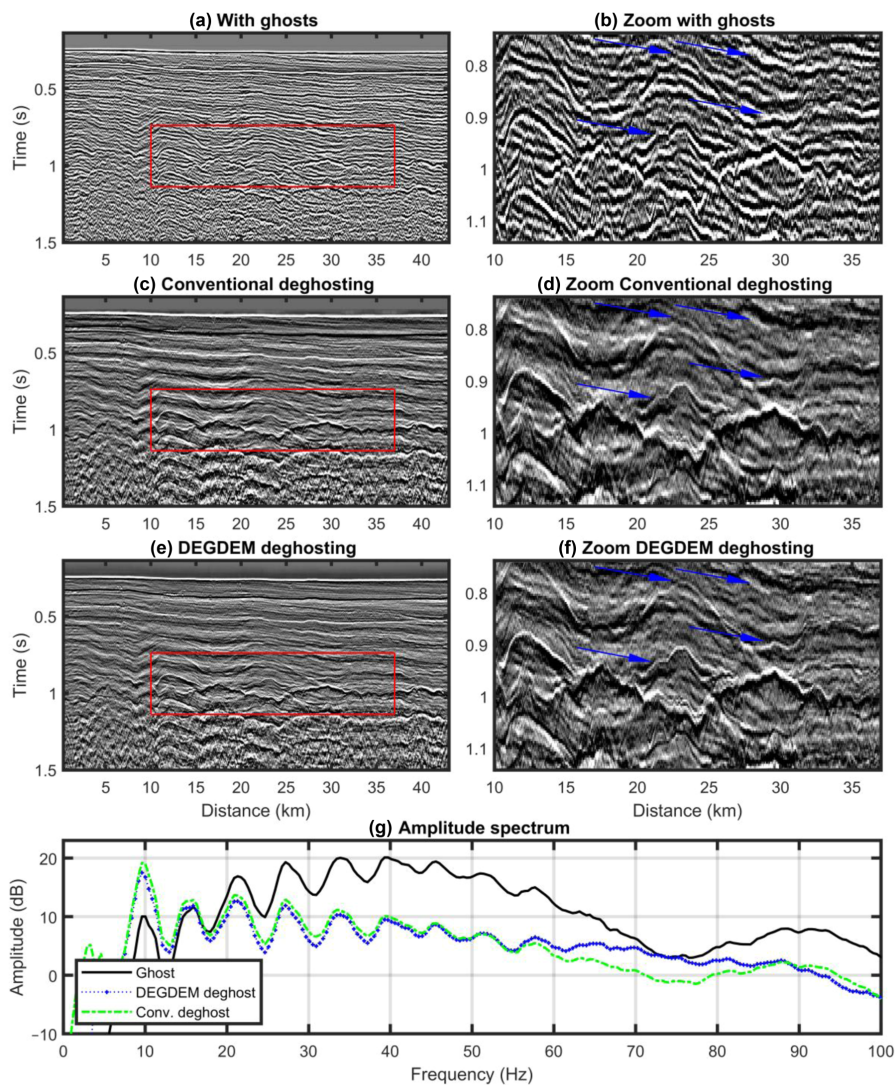
## DISCUSSION

### Synthetic data

The synthetic tests followed the same workflow as we would use on real data (excluding some processing steps such as deblending and designature because we modelled ‘clean’ finite-difference (FD) data without these types of noise). FD modelling was used to create the pre-stack depth migration (PSDM) image. Afterwards, we modelled training data with Kirchhoff demigration using the PSDM image. The network was then applied to the FD data. This approach avoided creating training data and prediction data using the same modelling approach, which could bias the results.

The ‘baseline test’ demonstrated that the neural network removed most of the ghosts and recovered amplitude in the notches (Figure 11). In addition, it also showed that the network can be trained on one part of the data and applied

successfully to another. The conventional deghosting worked better than DEGDEM (DEGghosting using DEMigration-based supervised learning) when source/receiver depth and sea surface reflectivity coefficient are perfectly known. DEGDEM may have performed better than conventional deghosting in the ghost notches because the training data were similar to the prediction data, which is one of the advantages when using demigration to create training data. In other words, it was already familiar with the ‘spiky’ nature of the prediction input. The conventional deghosting, on the other hand, was more general and did not have a bias towards any wavelet or geology. In addition, DEGDEM performed less well than the conventional deghosting on conflicting dipping events. This is believed to be because these conflicting dips were less well represented in the training data of the convolutional neural network (CNN). It could also be related to inaccurate demigration in the complicated areas because the migration result that leads to the PSDM image is poor in those areas. The demigration affects the training data in terms of similarity between the training data and prediction data. Another reason could be related to DEGDEM using the shot gather domain which might not be ideal for source deghosting.

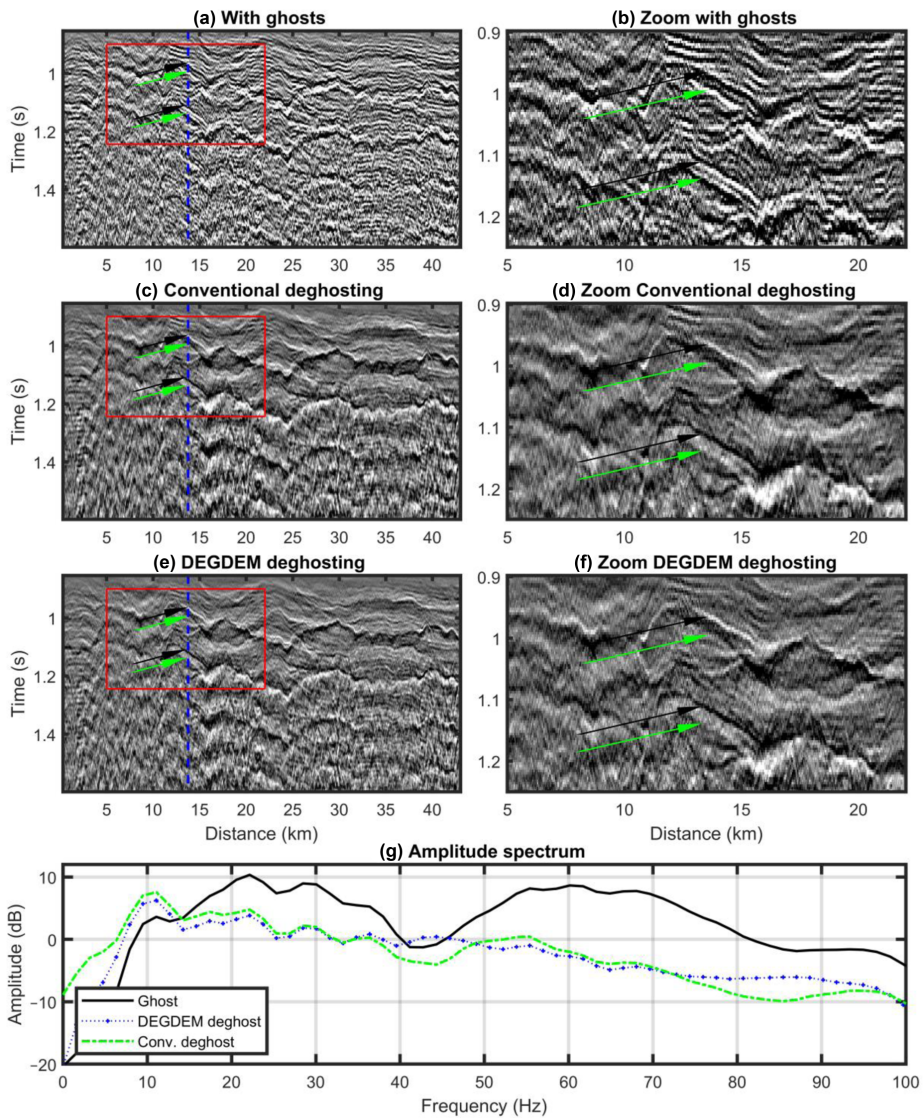


**FIGURE 21** Results on real data shown in the common channel gather (channel 15). (a) and (b) Data with ghosts. (c) and (d) Conventional deghosting. (e) and (f) DEGDEM deghosting. The red area in subfigures a, c and e is used as a close-up in subfigures b, d, and f. (g) Amplitude spectrum for data with ghosts (black), DEGDEM deghosting (blue) and conventional deghosting (green). The blue arrows indicate places where the resolution is increased.

Conventional source deghosting is often carried out on receiver gathers since the source ghost notch will be a function of apparent slowness in this domain, e.g., visible in the  $f$ - $k$  domain. However, the source ghost notch may be incorrect in the shot gather since the relation between the emission and incidence angles is not straightforward in a complex medium

(Blacquièrè & Sertlek, 2019). As a result, energy with one emission angle may arrive with another or multiple angles. Correspondingly, the receiver ghost notch is better defined in the shot gathers. Consequently, receiver deghosting is often done in shot gathers. Instead of doing both the source and receiver deghosting in one operation on shot gathers, we



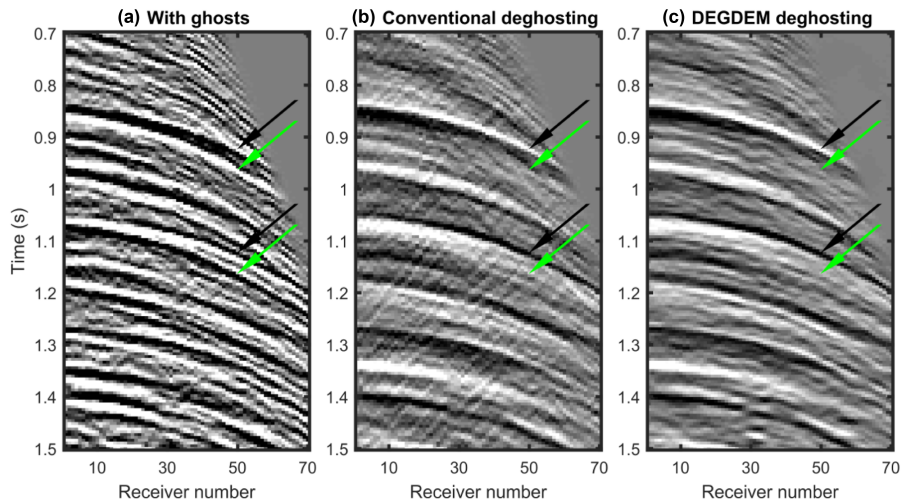


**FIGURE 22** Results on real data shown in the common channel gather (channel 50). (a) and (b) Data with ghosts. (c) and (d) Conventional deghosting. (e) and (f) DEGDEM deghosting. The red area in subfigures a, c, and e is used as a close-up in subfigures b, d, and f. (g) Amplitude spectrum for data with ghosts (black), DEGDEM deghosting (blue) and conventional deghosting (green). Black arrows indicate the primary or multiple, and green arrows indicate places with receiver ghosts. The vertical dashed blue line is the location of the shots shown in Figure 23.

could create a CNN for source deghosting on receiver gathers and another CNN for receiver deghosting on shot gathers. However, this is a topic for future research.

A clear conclusion of the synthetic tests is that DEGDEM was less sensitive to swells, variations in sea surface

reflection coefficient and errors in shot/receiver depths than the conventional method. An example was demonstrated on the data without and with swells where we observed a moderate change in the normalized root mean square (NRMS) error (0.0032–0.0044) using DEGDEM (U-net) compared to



**FIGURE 23** Results on real data on a shot gather. (a) Data with ghosts. (b) Conventional deghosting. (c) DEGDEM deghosting. Black arrows indicate the primary and multiple and green arrows indicate the timing of the receiver ghost. The shot location is shown in Figure 22 as a vertical dashed blue line.

the significant change in NRMS error (0.0030–0.0171) of the conventional method. It is possible to estimate the sea state (Orji et al., 2012) and use it to improve the deghosting (King & Poole, 2015; Vrolijk & Blacquièrè, 2018), but this adds more complexity to the deghosting process. Our results showed that the U-net can be made more robust to errors in streamer depths and sea surface reflection coefficients. This is potentially advantageous since there will always be some error when estimating these parameters on real data.

The test to investigate the sensitivity to residual ghost in the PSDM image showed, not surprisingly, that the networks performed less well than in the case where a fully ghost-free PSDM image was used. The DEGDEM (U-net) NRMS error using a clean PSDM image was 0.0044 versus 0.0070 when using a residual ghost PSDM image. However, the NRMS error was 0.0171 using the conventional deghosted data that went into the PSDM image. This suggests that DEGDEM can still improve on the quality of deghosting exhibited in the supplied PSDM image.

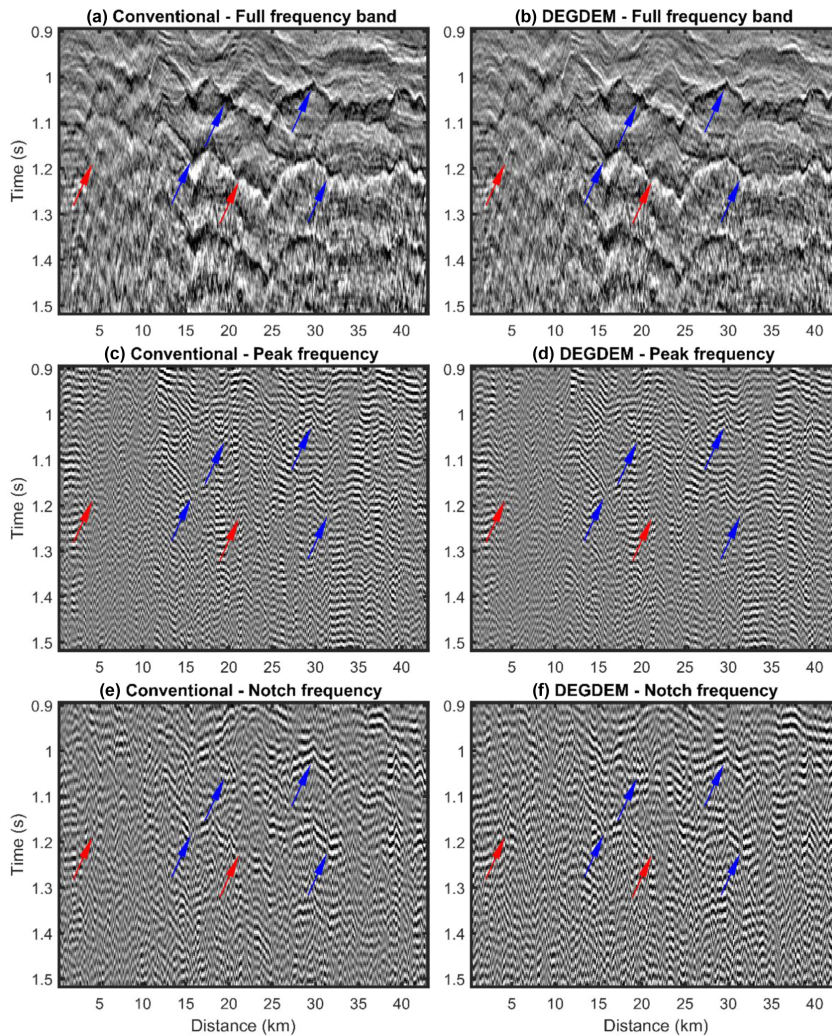
The final test demonstrated that the networks can deghost data with surface-related multiples (Figure 19) and preserve primaries interfering with multiples. This result is important since surface-related multiples are present in real data.

## Real data

Our results show that the network was able to deghost real three-dimensional (3D) data. Comparing our results with the

conventional method by Poole (2013) shows that the network resulted in higher amplitudes at the ghost notches. Figure 24 shows both conventional deghosting and DEGDEM at the second ghost peak ( $60 \pm 2$  Hz) and first ghost notch ( $42 \pm 2$  Hz). Both conventional and DEGDEM give similar results in the ghost peak (Figure 24c,d). Figure 24e,f does not clearly show that DEGDEM is able to recover a more coherent signal in this ghost notch. It is not easy to assess DEGDEM's ability to 'fill' the notch on real data. In addition, it is hard to assess the level of residual ghost energy in shallow water data due to the overlap of primary and reverberating multiple arrivals. We highlight an example of a possible residual ghost arrival, which was slightly weaker in the DEGDEM result compared to the conventional deghosting result.

In the real data example, we assumed a single-level source for both DEGDEM and conventional deghosting. Dual-level source deghosting was beyond the scope of our analysis mainly due to the added complexities it introduces for the demigration stage of the DEGDEM workflow. Nevertheless, Figure 25 shares some insight into what uplift may be provided by this approach. Figure 25 (left) shows results from joint source and receiver deghosting results assuming a single-level source as described in the previous section, and Figure 25 (right) shows results from joint source and receiver deghosting assuming a dual-level source (Poole et al., 2015). The image respecting the dual-level source setup looks sharper compared to the single-level source approach. It is possible that both the conventional and DEGDEM



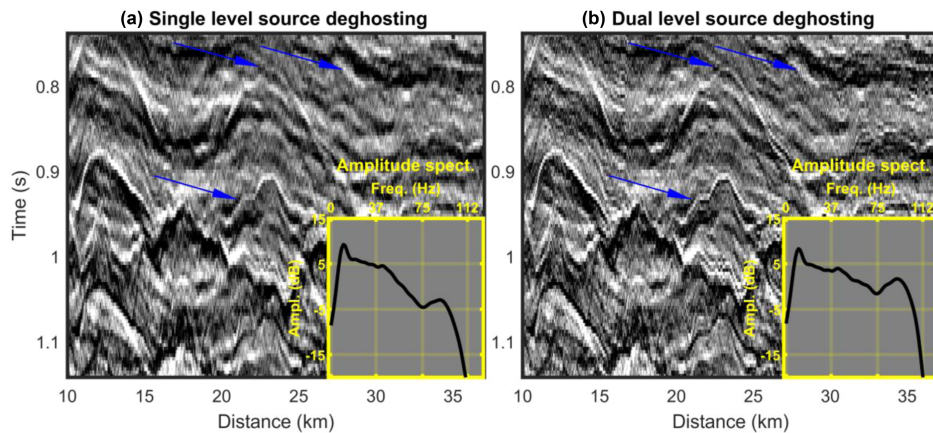
**FIGURE 24** (a) Conventional deghosting using the full spectrum, (b) DEGDEM deghosting using the full spectrum, (c) conventional deghosting at the peak frequency, (d) DEGDEM deghosting at the peak frequency, (E) conventional deghosting at the notch frequency and (F) DEGDEM deghosting at the notch frequency. Blue arrows indicate a coherent signal. Red arrows indicate what we interpret as noise or lack of coherent signal

deghosting could have given a better result if we had used a dual-level source.

Our results demonstrate that a network can be used on real data with good results. The main limitation of DEGDEM is the dependency on a PSDM image. This is similar to the ‘chicken-and-egg’ problem where we need to do deghosting to get a PSDM image but need the PSDM image to do deghosting. However, there are many such problems in seismic

processing. One example is the dependency on a PSDM image to create multiple models for demultiple, but demultiple data are needed to make the PSDM image (Brittan et al., 2011; Martin et al., 2011). Another example is the dependency on a velocity model for PSDM, but PSDM is dependency on a velocity model (Chang et al., 1996). A PSDM image is not always available, for example, in a new acquisition area or the early phase of a processing project. However, a PSDM image





**FIGURE 25** Conventional deghosting when we assume a single-level source (a) or a dual-level source (b). Blue arrows indicate better-defined reflectors

could be available from another acquisition in the same area or, as demonstrated here, from the same acquisition. Inaccuracy of migration velocity and limitation of PSDM could affect the performance of DEGDEM. It should also be noted that DEGDEM is computationally expensive since we need to create the demigrated training data, train a CNN, apply the CNN to the data and include an extra migration of the data. Assuming we already have a PSDM image, DEGDEM (U-net) is approximately 40 times slower when training and predicting on the same area compared to the conventional method used in this paper. For DEGDEM, this time includes demigration, training and prediction. The computation time for conventional deghosting and DEGDEM is highly dependent on the parameters and amount of training data used. However, it is possible that a network could be trained only on a representative part of the data and applied to the rest, similar to the method of de Jonge et al. (2021). Using that approach could eventually make DEGDEM faster than the conventional deghosting (assuming a PSDM image is already available). In principle, the basic idea behind the training approach in DEGDEM could be used for purposes other than deghosting, such as interpolation, demultiple, debubble, designation and more. However, investigating a similar workflow for these problems is a topic for future research.

## CONCLUSION

The results in this paper have demonstrated that DEGDEM (DEGghosting using DEMigration-based supervised learning) is capable of removing the source and receiver ghosts on both

synthetic and real data. On synthetic tests, we observe that the network was significantly less sensitive to errors in streamer depth and sea surface reflection coefficient compared to conventional deghosting.

On real data, DEGDEM showed a good level of energy in the ghost notch and provided locally better resolution compared to conventional deghosting. We examined a potential residual ghost arrival, which looked weaker on the DEGDEM result, but as discussed, it is hard to assess the level of residual ghost in shallow water datasets as many arrivals overlap. We demonstrated the robustness of DEGDEM to unknown changes in sea-state, receiver position and sea surface reflectivity, which are unavoidable on real data.


## ACKNOWLEDGEMENTS

The authors would like to thank CGG and the University of Bergen for providing the working environment for this research. We are grateful to CGG Multi-Client and Lundin for providing show rights. Further thanks go to Simon King, Karl Magnus Nielsen and the CGG internal reviewers, James Cooper and Ziqin Yu, for their helpful comments and suggestions that improved this work. We extend our gratitude to the Research Council of Norway, the University of Bergen and CGG for funding this work through an industrial Ph.D. grant, project no. 305450.

## DATA AVAILABILITY STATEMENT

The real data associated with this research are confidential and cannot be released. The Marmousi model can be found online: [https://wiki.seg.org/wiki/AGL\\_Elastic\\_Marmousi](https://wiki.seg.org/wiki/AGL_Elastic_Marmousi).

## ORCID

Thomas de Jonge  <https://orcid.org/0000-0001-7834-9526>

## REFERENCES

- Almuteri, K. & Sava, P. (2021) A convolutional neural network approach for ghost removal seismic deghosting using CNNs. In: *First International Meeting for Applied Geoscience & Energy Expanded Abstracts*, Society of Exploration Geophysicists, pp. 2550–2554.
- Amundsen, L. & Zhou, H. (2013) Low-frequency seismic deghosting. *Geophysics*, 78, WA15–WA20.
- Amundsen, L., Weglein, A.B. & Reitan, A. (2013a) On seismic deghosting using integral representation for the wave equation: Use of Green's functions with Neumann or Dirichlet boundary conditions. *Geophysics*, 78, T89–T98.
- Amundsen, L., Zhou, H., Reitan, A. & Weglein, A.B. (2013b) On seismic deghosting by spatial deconvolution. *Geophysics*, 78, 267–271.
- Asgedom, E.G., Cecconello, E., Orji, O.C. & Söllner, W. (2017) *Rough sea surface reflection coefficient estimation and its implication on hydrophone-only pre-stack Deghosting*. In: 79th EAGE Conference and Exhibition 2017. EAGE, pp. 12–15.
- Aytun, K. (1999) The footsteps of the receiver ghost in the f-k domain. *Geophysics*, 64, 1618–1626.
- Bearnth, R.E. & Moore, N.A. (1989) *Air gun-slant cable seismic results in the Gulf of Mexico*. In: 1989 SEG Annual Meeting. SEG, pp. 649–652.
- Blacquièrre, G. & Sertlek, H.Ö. (2019) Modeling and assessing the effects of the sea surface, from being flat to being rough and dynamic. *Geophysics*, 84, T13–T27.
- Brittan, J., Martin, T., Bekara, M. & Koch, K. (2011) 3D shallow water demultiple – extending the concept. *First Break*, 29, 97–101.
- Carlson, D., Long, A., Söllner, W., Tabti, H., Tengham, R. & Lunde, N. (2007) Increased resolution and penetration from a towed dual-sensor streamer. *First Break*, 25, 71–77.
- CGG. (2020) NVG 3D seismic. Available at <https://www.cgg.com/multi-client-data/multi-client-seismic/northern-viking-graben> (Accessed: 01 August 2022).
- Chang, H., Solano, M., VanDyke, J.P., McMechan, G.A. & Epili, D. (1996) 3-D prestack Kirchhoff depth migration: From prototype to production in a MPP environment. *Geophysics*, 63, 546–556.
- Cunha, A., Pochet, A., Lopes, H. & Gattass, M. (2020) Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data. *Computers and Geosciences*, 135, 1–9.
- Dumoulin, V. & Visin, F. (2016) A guide to convolution arithmetic for deep learning [preprint]. ArXiv, 1603.07285.
- Fang, W., Fu, L., Zhang, M. & Li, Z. (2021) Seismic data interpolation based on U-net with texture loss. *Geophysics*, 86, V41–V54.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016) *Deep Learning*. MIT Press.
- Greiner, T.L., Kolbjørnsen, O., Lie, J.E., Nilsen, E.H., Evensen, A.K. & Gelius, L. (2019) *Cross-streamer wavefield interpolation using deep convolutional neural network*. In: SEG Technical Program Expanded Abstracts 2019. SEG, pp. 2207–2211.
- Hill, D., Combee, C. & Bacon, J. (2006) Over/under acquisition and data processing: The next quantum leap in seismic technology? *First Break*, 24, 81–96.
- Hubral, P., Schleicher, J. & Tygel, M. (1996) A unified approach to 3-D seismic reflection imaging, Part I: Basic concepts. *Geophysics*, 61, 742–758.
- de Jonge, T., Vinje, V., Poole, G., Hou, S. & Iversen, E. (2021) De-bubbling Seismic Data using a Generalized Neural Network. *Geophysics*, 87, V1–V14.
- Jovanovich, D.B., Sumner, R.D. & Akins-Easterlin, S.L. (1983) Ghosting and marine signature deconvolution: A prerequisite for detailed seismic interpretation. *Geophysics*, 48, 1468–1485.
- King, S. & Poole, G. (2015) Hydrophone-only receiver deghosting using a variable sea surface datum. In: *SEG Technical Program Expanded Abstracts*. SEG, pp. 4610–4614.
- Klochikhina, E., Crawley, S., Frolov, S., Chemingui, N. & Martin, T. (2020) Leveraging deep learning for seismic image denoising. *First Break*, 38, 41–48.
- Li, J., Wang, B., Han, D. & Wang, Y. (2020) *Intelligent seismic deblending based deep learning based U-net*. In: 82nd EAGE Annual Conference and Exhibition, 2020. EAGE, pp. 1–5.
- Lucas, A., Iliadis, M., Molina, R. & Katsaggelos, A.K. (2018) Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35, 20–36.
- Martin, G.S., Wiley, R. & Marfurt, K.J. (2006) Marmousi2: An elastic upgrade for Marmousi. *Leading Edge (Tulsa, OK)*, 25, 156–166.
- Martin, T., Brittan, J., Bekara, M. & Koch, K. (2011) 3D shallow water demultiple - Extending the concept. In: 73rd European Association of Geoscientists and Engineers Conference and Exhibition 2011: Unconventional Resources and the Role of Technology. *Incorporating SPE EUROPEC 2011*, 3. European Association of Geoscientists and Engineers, pp. 2040–2045.
- Mellier, G. & Tellier, N. (2018) *Considerations about multi-sensor solid streamer design*. In: EAGE Marine Acquisition Workshop, Oslo, Norway. EAGE, cp-560-0000.7.
- Orji, O.C., Söllner, W. & Gelius, L.J. (2012) Effects of time-varying sea surface in marine seismic data. *Geophysics*, 77, P33–P43.
- Orji, O.C., Sollner, W. & Gelius, L.J. (2013) Sea surface reflection coefficient estimation. In: Society of Exploration Geophysicists International Exposition and 83rd Annual Meeting, SEG 2013. *Expanding Geophysical Frontiers*. SEG, pp. 51–55.
- Peng, C., Jin, H. & Wang, P. (2014) *Noise attenuation for multi-sensor streamer data via cooperative de-noising*. In: Society of Exploration Geophysicists International Exposition and 84th Annual Meeting SEG 2014. SEG, pp. 1878–1882.
- Peng, H., Messud, J., Salaun, N., Hammoud, I., Jeunesse, P., Lesieur, T. & Lacombe, C. (2021) *Proposal of the Dunet neural network architecture: deghosting example and theoretical analysis*. In: 82nd EAGE Annual Conference & Exhibition. EAGE, pp. 1–5.
- Poole, G. (2013) Pre-migration receiver de-ghosting and re-datuming for variable depth streamer data In: Society of Exploration Geophysicists International Exposition and 83rd Annual Meeting, SEG 2013. *Expanding Geophysical Frontiers*. SEG, pp. 4216–4220.
- Poole, G. & Cooper, J. (2018) *Multi-sensor receiver deghosting using data domain sparseness weights*. In: 80th EAGE Conference and Exhibition 2018: Opportunities Presented by the Energy Transition. EAGE, pp. 1–5.
- Poole, G., Cooper, J., King, S. & Wang, P. (2015) *3D source signature using source–receiver symmetry in the shot tau-px-py domain*. In: 77th EAGE Conference and Exhibition 2015: Earth Science for Energy and Environment. EAGE, pp. 3867–3871.
- Qu, S., Verschuur, E., Zhang, D. & Chen, Y. (2021) Training deep networks with only synthetic data: deep-learning-based near-offset reconstruction for (closed-loop) surface-related multiple estimations on shallow-water field data. *Geophysics*, 86, A39–A43.



- Rickett, J.E., van Manen, D.J., Loganathan, P. & Seymour, N. (2014) *Slanted-streamer data-adaptive deghosting with local plane waves*. In: *76th EAGE Conference and Exhibition 2014*. EAGE, pp. 1–5.
- Ronneberger, O., Fischer, P. & Brox, T. (2015) *U-net: convolutional networks for biomedical image segmentation*. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 234–241.
- Santos, L. T., Schleicher, J., Tygel, M. & Hubral, P. (2000a) Seismic modeling by demigration. *Geophysics*, 65, 1281–1289.
- Santos, L. T., Schleicher, J., Tygel, M. & Hubral, P. (2000b) Modeling, migration, and demigration. *Leading Edge (Tulsa, OK)*, 19, 712–715.
- Schuster, G.T. (1993) Least-squares cross-well migration. In: *SEG Technical Program Expanded Abstracts*. SEG, pp. 110–113.
- Siahkoochi, A., Kumar, R. & Herrmann, F. (2018) *Seismic data reconstruction with generative adversarial networks*. In: *80th EAGE Conference and Exhibition 2018: Opportunities Presented by the Energy Transition, 2018*. EAGE, pp. 1–5.
- Siliqi, R., Payen, T., Sablon, R. & Desruets, K. (2013) Synchronized multi-level source, a robust broadband marine solution. In: *SEG Technical Program Expanded Abstracts 2013*. Society of Exploration Geophysicists, pp. 56–60.
- Song, J.G., Gong, Y.L. & Li, S. (2015) High-resolution frequency-domain Radon transform and variable-depth streamer data deghosting. *Applied Geophysics*, 12, 564–572.
- Soubaras, R. (2010) *Deghosting by joint deconvolution of a migration and a mirror migration*. In: *Annual Meeting SEG Denver 2010 Annual Meeting*. SEG, pp. 3406–3410.
- Soubaras, R. & Dowle, R. (2010) Variable-depth streamer - a broadband marine solution. *First Break*, 28, 89–96.
- Soubaras, R., Dowle, R. & Sablon, R. (2012) BroadSeis: Enhancing interpretation and inversion with broadband marine seismic.: *CSEG Recorder*, 37, 41–46.
- Sun, J., Slang, S., Elboth, T., Greiner, T. L., McDonald, S. & [J] Gelius, L. (2019) Attenuation of marine seismic interference noise employing a customized U-Net. *Geophysical Prospecting*, 68, 845–871.
- Vrolijk, J.-W. & Blacquièrre, G. (2021) Source deghosting of coarsely sampled common-receiver data using a convolutional neural network. *Geophysics*, 86, V185–V196.
- Vrolijk, J. & Blacquièrre, G. (2018) Adaptive deghosting including the rough and time variant sea surface. In: *80th EAGE Conference and Exhibition 2018: Opportunities Presented by the Energy Transition*. EAGE, pp. 1–5.
- Vrolijk, J.W. & Blacquièrre, G. (2020) Source deghosting of coarsely sampled common-receiver data using machine learning. In: *SEG Technical Program Expanded Abstracts*. SEG, pp. 3294–3298.
- Zhang, Z., Masoomzadeh, H. & Wang, B. (2018) Evolution of deghosting process for single-sensor streamer data from 2D to 3D. *Geophysical Prospecting*, 66, 975–986.
- Zu, S., Cao, J., Qu, S. & Chen, Y. (2020) Iterative deblending for simultaneous source data using the deep neural network. *Geophysics*, 85, V131–V141.

**How to cite this article:** de Jonge, T., Vinje, V., Zhao, P., Poole, G. & Iversen, E. (2022) Source and receiver deghosting by demigration-based supervised learning. *Geophysical Prospecting*, 1–26. <https://doi.org/10.1111/1365-2478.13253>

## 6.3 Article III

Title: *Dehosting dual-component streamer data using demigration-based supervised learning*

Authors: Thomas de Jonge, Vetle Vinje, Gordon Poole, Peng Zhao, and Einar Iversen

Submitted to review in *Geophysical Prospecting* (2022).

Submitted version reprinted. Reprint published under Open Access license CC-BY 4.0.

Copyright: The Authors 2023.



# Chapter 7

## Conclusion, discussion, and future work

In this chapter, I mention the key findings from each paper and put them into a common context. Then, I briefly discuss why my research is important and how it relates to similar research. Finally, I discuss possible future research directions.

### 7.1 Key findings

The key findings from each paper can be summarized as follows:

- In paper 1, we trained a network on real data containing a large range of source signatures. This diversity made the network robust and adaptive to signature variations which is an advantage if the signature change during an acquisition. The network performs well if the signature in the test data is equal to one in the training data. We also investigated the network sensitivity to geology within an extensive survey and two surveys on the Norwegian Continental Shelf (NCS). If the test data are from geology similar to the training data, the network performs better than if not. However, even when we applied the network to a different part of the NCS, the network could still reduce most of the bubble noise.
- In paper 2, we created training data with and without ghosts using demigration of a stacked depth-migrated image. We showed on synthetic data that a convolutional neural network was more robust than a conventional deghosting method to variability in source/receiver depth, swells, and sea surface reflectivity. In addition, on synthetic data, we demonstrated that it is possible to make a network

more robust to, e.g., swells by perturbing the receiver depth in the training data. On real North Sea data, in some respects, the neural network performed better than a standard deterministic deghosting method which was based on inversion in the  $\tau - p$  domain.

- In paper 3, we expand upon the method in paper 2 to include dual-component data in training and prediction. The dual-component training data is created using demigration of a stacked depth-migrated image into vertical particle velocity data and pressure data. We test our method on synthetic Marmousi and real North Sea data with dual-component streamers. We compare the method with conventional dual-component deghosting using pressure and vertical velocity summation. The synthetic results show that the method can accurately remove the ghosts with only minor errors. Our method is less sensitive to spatially aliased data than the conventional method. On real data, the neural network deghosting show consistency with conventional deghosting, both within and outside the training area.

Papers 2 and 3 are closely related because both aim to remove ghosts. Paper 1 aims to remove the bubble noise. These are two important processing steps done after a conventional marine seismic acquisition.

Paper 1 investigates creating a generalized neural network that can handle different source signatures. Paper 2 investigates how a generalized network can help with deghosting. In paper 2, we made the network robust to receiver depth and sea surface reflection coefficient changes. We did not investigate generalization in the third paper, but the generalization research done in the second paper is likely valid for the work done in paper 3. The ghost model and source signature can change during a seismic survey. Therefore, the network's ability to generalize and adapt to changes in the prediction data is an attractive feature. It can allow less testing and quality control time compared to conventional methods.

Paper 1 demonstrated that the network is somewhat sensitive to the geology over a large survey and for two different surveys. However, even when we applied the network to a survey on a different part of the Norwegian Continental Shelf, the network could remove most of the bubble noise. We did not attempt to apply a trained network to another part of the NCS in papers 2 or 3 for deghosting. However, in these two papers, we used only part of a sail line as the "training area" (the training area is the area from the pre-stack depth migration (PSDM) image that is used to create training data). However, when we apply the network to the full sail line, the deghosting quality seems to be the same. Based on the results from papers 1, 2, and 3, a network works well, at least

within the same survey, if the geology does not change too much.

The papers show that the network can be used readily without much testing. In all papers, we always compare with the conventional methods that professional geophysicists have extensively tested. This means we compare high-quality results acquired during, in some cases, weeks or months of testing. Even though our results show good quality, it is possible to use pre-trained networks for debubbling or deghosting quickly after seismic surveys as a fast track. It is also possible to use a trained network from another survey and data from the new survey to fine-tune the network.

## 7.2 Discussion

All three papers focused on how training data could be used to solve the debubbling or deghosting problem. While the structure of a neural network is important to acquire good results, extensive research to develop neural networks has already been done, and plenty of resources can be found easily. However, creating training data for debubbling and deghosting is an immature field. People have created training data using different approaches in other related neural network processing steps. Some papers use pure synthetic data as training data [Qu et al., 2021; Zu et al., 2020]. Other papers train a network on synthetic data and use real data for fine-tuning [Cunha et al., 2020; Li et al., 2021]. Another option is to utilize similarities between training and prediction data in two different domains [Greiner et al., 2019; Siahkoochi et al., 2018; Vrolijk and Blacquièrè, 2021]. Some papers use conventional methods to create real training data [Peng et al., 2021]. In paper 1, we "add" synthetic bubble noise to real debubbled data. Therefore, this training data is not fully synthetic or fully real. In papers 2 and 3, we create training data using demigration that looks highly realistic and is arguably not fully synthetic. Regardless, we create training data containing complex features similar to real data, which is an advantage.

The main limitation in the second and third papers is that we need a PSDM image to create training data. This limitation is similar to the "chicken or the egg" dilemma, where we need to remove the ghost before we create the PSDM image but need the PSDM image for deghosting. In paper 2 and paper 3, we solved this problem by using a conventional deghosting method before creating a PSDM image. However, there are other seismic processing examples with this problem. Some demultiple methods require a PSDM image to create multiple models, but demultiple is required to create a PSDM image [Brittan et al., 2011; Martin et al., 2011]. Also, some velocity model building

methods require a PSDM image, but the PSDM image requires a velocity model [Chang et al., 1998]. However, in a standard modern processing project, the processing from the raw data to the PSDM image is repeated iteratively to create better and better images. Therefore, a PSDM image could be available early in a processing project. In addition, as we have seen from papers 2 and 3, the PSDM image from the full survey area is not needed to create training data. Therefore, one could create a PSDM image of a smaller area and predict the full survey. There are many possibilities, and the method from papers 2 and 3 could be a nice tool among other deghosting methods.

A pre-trained neural network for deghosting or debubbling could be used quickly on newly acquired seismic data. The pre-trained network could be trained on, e.g., a neighboring area with an available PSDM image. Alternatively, a pre-trained network from recently acquired seismic data could be used on extensions of the original survey, e.g., where a new extension is done each season. Pre-trained networks for different processing tasks open the possibility of quickly processing raw data to the final PSDM image with good results.

### 7.3 Future work

There are several interesting future research paths in all three papers. In the first paper, I perturbed the source signature to make the network robust to changes in bubble noise. The perturbation was done by changing the phase and amplitude of one or two source signatures. However, more sophisticated methods could create a larger range of realistic source signatures. An option would be to model source signatures believed to be found in the data. Perhaps, a more sophisticated method of sampling these signatures (e.g., pseudo-randomly) could be done.

It could also be possible to use time migration to create a reflectivity image instead of depth migration. Time demigration would then be used to create training data. Creating images from time migration is less computationally expensive than depth migration. Also, time-domain images are usually available relatively early in the processing sequence [Iversen et al., 2012]. Therefore, time-domain images may be available earlier and we can create training data sooner at a reduced computational cost.

Kirchhoff depth migration is generally sensitive to the velocity model. However, since we are doing both migration and demigration, the dependency on the velocity model could be eliminated or reduced since these are reverse processes. Therefore, having a

well-defined velocity model might not be important before creating training data through demigration. This thesis did not investigate the importance of the velocity model for migration and demigration. However, this would be interesting for future research.

In paper 2, we created a PSDM image from data with residual ghost noise. Consequently, the demigrated training data was created from this PSDM containing residual ghost noise. However, the network could deghost the data better than the data used to create the PSDM image. This indicates that it could be possible to update the PSDM image to improve the deghosting iteratively. This process would be quite expensive when using depth migration and demigration. However, time migration and demigration could be a better option.

Demigration could also be used to create training data for other processing steps. The paper by Hlebnikov et al. [2022] has already shown that demigration-based supervised learning can be used for offset class interpolation. Most likely, demigration-based supervised learning could be used to interpolate data close to zero-offset, between shots, or shot lines. Perhaps it is possible to train a network to remove multiples by creating demigrated training data with and without multiples. Another possibility could be to remove swell noise by creating demigrated data with and without swell. Swell noise could be extracted from real data without any signal (e.g., at the end of the sail line) and added to the demigrated data. Perhaps, demigration could also be used to denoise Vz data using a similar approach. These options are only a few possible ways to use demigration-based supervised learning for processing seismic data.





# Bibliography

- L. Amundsen, A. B. Weglein, and A. Reitan. On seismic deghosting using integral representation for the wave equation: Use of green's functions with neumann or dirichlet boundary conditions. *Geophysics*, 78(4), 2013a. doi: 10.1190/GEO2012-0305.1.
- L. Amundsen, H. Zhou, A. Reitan, and A. B. Weglein. On seismic deghosting by spatial deconvolution. *Geophysics*, 78(6):267–271, 2013b. doi: 10.1190/GEO2013-0198.1.
- R. E. Bearnth and N. A. Moore. Air gun-slant cable seismic results in the gulf of mexico. *59th Annual International Meeting, SEG, Expanded Abstracts*, pages 649–652, 1989. doi: 10.1190/1.1889556.
- N. Bleistein. On the imaging of reflectors in the earth. *Geophysics*, 52(7):931–942, 1987. doi: 10.1190/1.1442363.
- J. Brittan, T. Martin, M. Bekara, and K. Koch. 3d shallow water demultiple – extending the concept. *First Break*, 29(9):97–101, 2011. doi: 10.3997/1365-2397.29.9.53730.
- D. Carlson, A. Long, W. Söllner, H. Tabti, R. TENGHAMN, and N. Lunde. Increased resolution and penetration from a towed dual-sensor streamer. *First Break*, 25(12): 71–77, 2007. doi: 10.3997/1365-2397.25.12.27722.
- H. Chang, J. P. VanDyke, M. Solano, G. A. McMechan, and D. Epili. 3-d prestack kirchhoff depth migration: From prototype to production in a massively parallel processor environment. *Geophysics*, 63(2):546–556, 1998. doi: 10.1190/1.1444355.
- A. Cunha, A. Pochet, H. Lopes, and M. Gattass. Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data. *Computers & Geosciences*, 135:1–9, 2020. doi: 10.1016/j.cageo.2019.104344.
- C. M. Davison and G. Poole. Far-field source signature reconstruction using direct arrival data. *77th Annual International Conference and Exhibition, EAGE, Extended Abstracts*, pages 4472–4476, 2015. doi: 10.3997/2214-4609.201413326.

- T. de Jonge, V. Vinje, P. Poole, Gordon Zhao, and E. Iversen. Deghosting dual-component streamer data using demigration-based supervised learning. *Submitted to review in Geophysical Prospecting*, 2022a.
- T. de Jonge, V. Vinje, G. Poole, S. Hou, and E. Iversen. De-bubbling seismic data using a generalized neural network. *Geophysics*, 87(1):V1–V14, 2022b. doi: 10.1190/geo2021-0053.1.
- T. de Jonge, V. Vinje, P. Zhao, G. Poole, and E. Iversen. Source and receiver deghosting by demigration-based supervised learning. *Geophysical Prospecting*, pages 1–26, 2022c. doi: 10.1111/1365-2478.13253.
- P. Domingos. A few useful things to know about machine learning. *Commun. ACM*, 55(10):78–87, 2012. doi: 10.1145/2347736.2347755.
- B. Dragoset. Introduction to air guns and air-gun arrays. *Leading Edge*, 19(8):892–897, 2000. doi: 10.1190/1.1438741.
- V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *ArXiv*, 1603.07285, 2016.
- J. T. Etgen and C. Kumar. What really is the difference between time and depth migration? a tutorial. *82nd Annual International Meeting, SEG, Expanded Abstracts*, pages 1–5, 2012. doi: 10.1190/segam2012-0266.1.
- I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
- T. L. Greiner, O. Kolbjørnsen, J. E. Lie, E. H. Nilsen, A. K. Evensen, and L. Gelius. Cross-streamer wavefield interpolation using deep convolutional neural network. *89th Annual International Meeting, SEG, Expanded Abstracts*, pages 2207–2211, 2019. doi: 10.1190/segam2019-3214009.1.
- S. Grion, G. Williams, R. Light, S. Denny, and R. Telling. Signature estimation and drop-out implications for a triple source marine seismic survey. *80th Annual International Conference and Exhibition, EAGE, Extended Abstracts*, pages 1–5, 2018. doi: 10.3997/2214-4609.201800742.
- J. W. Hammond. Ghost elimination from reflection records. *Geophysics*, 27(1):48–60, 1962. doi: 10.1190/1.1438977.
- D. Hill, L. Combee, and J. Bacon. Over/under acquisition and data processing: The next quantum leap in seismic technology? *First Break*, 24(6):81–95, 2006. doi: 10.3997/1365-2397.24.1096.26991.

- V. Hlebnikov, T. Elboth, V. Vinje, and L.-J. Gelius. Noise types and their attenuation in towed marine seismic: A tutorial. *Geophysics*, 86(2):W1–W19, 2021. doi: 10.1190/geo2019-0808.1.
- V. Hlebnikov, T. A. L. Greiner, V. Vinje, J. E. Lie, and G. Poole. De-migration-based supervised learning for interpolation and regularization of 3d offset classes. *Geophysical Prospecting*, pages 1–17, 2022. doi: 10.1111/1365-2478.13206.
- E. Iversen, M. Tygel, B. Ursin, and M. V. de Hoop. Kinematic time migration and demigration of reflections in pre-stack seismic data. *Geophysical Journal International*, 189(3):1635–1666, 2012. doi: 10.1111/j.1365-246X.2012.05435.x.
- D. T. Johnson. Understanding air-gun bubble behavior. *Geophysics*, 59(11):1173–1729, 1994. doi: 10.1190/1.1443559.
- D. B. Jovanovich, R. D. Sumner, and S. L. Akins-Easterlin. Ghosting and marine signature deconvolution: a prerequisite for detailed seismic interpretation. *Geophysics*, 48(11):1468–1485, 1983. doi: 10.1190/1.1441431.
- M. Landrø. Modelling of gi gun signatures. *Geophysical Prospecting*, 40(7):721–747, 1992. doi: 10.1111/j.1365-2478.1992.tb00549.x.
- M. Landrø and L. Amundsen. Marine seismic sources part i. *Geo ExPro*, 7(1):32–34, 2010.
- Learning Geology - A Geology Outreach Website. Marine and land seismic acquisition, 2015. URL <http://geologylearn.blogspot.com/2015/06/marine-and-land-seismic-aquisition.html>. Accessed: 2022-27-12.
- J. Li, B. Wang, D. Han, and Y. Wang. Intelligent seismic deblending based deep learning based u-net. *82nd Annual International Conference and Exhibition, EAGE, Extended Abstracts*, pages 1–5, 2021. doi: 10.3997/2214-4609.202010691.
- J. P. Lindsey. Elimination of seismic ghost reflections by means of a linear filter. *Geophysics*, 25(1):130–140, 1960. doi: 10.1190/1.1438679.
- L. Lines, A. Burton, and H. xing Lu. Optimization methods for prestack and poststack migration (migration without migraines). *CSEG Recorder*, 19(4):1–11, 1994.
- A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018. doi: 10.1109/MSP.2017.2760358.
- G. S. Martin, R. Wiley, and K. J. Marfurt. Marmousi2: An elastic upgrade for marmousi. *Leading Edge*, 25(2):156–166, 2006. doi: 10.1190/1.2172306.

- T. Martin, J. Brittan, M. Bekara, and K. Koch. 3d shallow water demultiple - extending the concept. *73rd Annual International Conference and Exhibition, EAGE, Extended Abstracts*, pages 2040–2045, 2011. doi: 10.3997/2214-4609.20149267.
- G. Mellier and N. Tellier. Considerations about multi-sensor solid streamer design. *EAGE Marine Acquisition Workshop, Oslo, Norway*, Th MA 01, 2018. doi: 10.3997/2214-4609.201802092.
- Y. Ni, C. Niang, and R. Siliqi. Monitoring the stability of airgun source array signature. *82nd Annual International Meeting, SEG, Expanded Abstracts*, pages 214–218, 2012. doi: 10.1190/segam2012-0875.1.
- Norwegian Petroleum. Exports of oil and gas, 2022. URL <https://www.norskipetroleum.no/produksjon-og-eksport/eksport-av-olje-og-gass/>. Accessed: 2022-27-12.
- Oljedirektoratet. Faktaboks 4.1 borekostnader, 2020. URL <https://www.npd.no/fakta/publikasjoner/rapporter/ressursrapporter/ressursrapport-2020/betydning-av-leting/faktaboks-4-1-borekostnader/>. Accessed: 2022-27-12.
- O. C. Orji, W. Sollner, and L. J. Gelius. Sea surface reflection coefficient estimation. *83rd Annual International Meeting, SEG, Expanded Abstracts*, pages 51–55, 2013. doi: 10.1190/segam2013-0944.1.
- C. Peng, H. Jin, and P. Wang. Noise attenuation for multi-sensor streamer data via cooperative de-noising. *84th Annual International Meeting, SEG, Expanded Abstracts*, pages 1878–1882, 2014. doi: 10.1190/segam2014-0300.1.
- H. Peng, J. Messud, N. Salaun, I. Hammoud, P. Jeunesse, T. Lesieur, and C. Lacombe. Proposal of the dunet neural network architecture: Deghosting example and theoretical analysis. *82nd Annual International Conference and Exhibition, EAGE, Extended Abstracts*, pages 1–5, 2021. doi: 10.3997/2214-4609.202112820.
- G. Poole. Pre-migration receiver de-ghosting and re-datuming for variable depth streamer data. *83rd Annual International Meeting, SEG, Expanded Abstracts*, pages 4216–4220, 2013. doi: 10.1190/segam2013-0541.1.
- G. Poole and J. Cooper. Multi-sensor receiver deghosting using data domain sparseness weights. *80th Annual International Conference and Exhibition, EAGE, Extended Abstracts*, We A10 08, 2018. doi: 10.3997/2214-4609.201801003.
- S. Qu, E. Verschuur, D. Zhang, and Y. Chen. Training deep networks with only synthetic data: Deep-learning-based near-offset reconstruction for (closed-loop) surface-related

- multiple estimation on shallow-water field data. *Geophysics*, 86(3):A39–A43, 2021. doi: 10.1190/geo2020-0723.1.
- J. Rickett, D. van Manen, P. Loganathan, and N. Seymour. Slanted-streamer data-adaptive deghosting with local plane waves. *76th Annual International Conference and Exhibition, EAGE, Extended Abstracts*, pages 1–5, 2014. doi: 10.3997/2214-4609.20141453.
- H. Ritchie, M. Roser, and P. Rosado. Energy. *Our World in Data*, 2022. URL <https://ourworldindata.org/energy>.
- O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.
- L. T. Santos, J. Schleicher, M. Tygel, and P. Hubral. Modeling, migration, and demigration. *Leading Edge*, 19(7):712–715, 2000a. doi: 10.1190/1.1438696.
- L. T. Santos, J. Schleicher, M. Tygel, and P. Hubral. Seismic modeling by demigration. *Geophysics*, 65(4):1281–1289, 2000b. doi: 10.1190/1.1444819.
- W. A. Schneider, K. L. Lerner, J. P. Burg, and M. M. Backus. A new data-processing technique for the elimination of ghost arrivals on reflection seismograms. *Geophysics*, 29(5):783–805, 1964. doi: 10.1190/1.1439419.
- P. Scholtz, H. Masoomzadeh, and R. Camp. Directional signature without near-field hydrophone recordings. *85th Annual International Meeting, SEG, Expanded Abstracts*, pages 4423–4427, 2015. doi: 10.1190/segam2015-5843268.1.
- A. Siahkoochi, R. Kumar, and F. Herrmann. Seismic data reconstruction with generative adversarial networks. *80th Annual International Conference and Exhibition, EAGE, Extended Abstracts*, pages 1–5, 2018. doi: 10.3997/2214-4609.201801393.
- R. Soubaras. Deghosting by joint deconvolution of a migration and a mirror migration. *80th Annual International Meeting, SEG, Expanded Abstracts*, pages 3406–3410, 2010. doi: 10.1190/1.3513556.
- R. Soubaras and R. Dowle. Variable-depth streamer - a broadband marine solution. *First Break*, 28(12):89–96, 2010. doi: 10.3997/1365-2397.28.12.44692.
- R. Soubaras, R. Dowle, and R. Sablon. Variable depth streamer acquisition: Enhancing interpretation with broadband marine seismic. *CSEG Recorder*, 37(7):40–46, 2012.

- R. Telling and S. Grion. Multicomponent de-ghosting using a hybrid operator in frequency and space. *83rd Annual International Conference and Exhibition, EAGE, Extended Abstracts*, pages 1–5, 2022. doi: 10.3997/2214-4609.202210526.
- R. Tenghamn and P. E. Dhelie. Geostreamer - increasing the signal-to-noise ratio using a dual-sensor towed streamer. *First Break*, 27(10):45–51, 2009. doi: 10.3997/1365-2397.2009017.
- M. Tygel, J. Schleichert, and P. Hubral. A unified approach to 3-d seismic reflection imaging, part ii: Theory. *Geophysics*, 61(3):759–775, 1996. doi: 10.1190/1.1444001.
- J.-W. Vrolijk and G. Blacquière. Source deghosting of coarsely sampled common-receiver data using a convolutional neural network. *Geophysics*, 86(3):V185–V196, 2021. doi: 10.1190/geo2020-0186.1.
- P. Wang, S. Ray, C. Peng, Y. Li, and G. Poole. Premigration deghosting for marine streamer data using a bootstrap approach in tau-p domain. *83rd Annual International Meeting, SEG, Expanded Abstracts*, pages 4221–4225, 2013. doi: 10.1190/segam2013-0225.1.
- Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.
- Ö. Yilmaz. *Seismic data analysis: Processing, inversion, and interpretation of seismic data*. Society of exploration geophysicists, 2001.
- Z. Zhang, H. Masoomzadeh, and B. Wang. Evolution of deghosting process for single-sensor streamer data from 2d to 3d. *Geophysical Prospecting*, 66(5):975–986, 2018. doi: 10.1111/1365-2478.12614.
- A. Ziolkowski. A method for calculating the output pressure waveform from an air gun. *Geophysical Journal International*, 21(2):137–161, 1970. doi: 10.1111/j.1365-246X.1970.tb01773.x.
- A. Ziolkowski and R. G. K. Johnston. Marine seismic sources: Qc of wavefield computation from near-field pressure measurements. *Geophysical Prospecting*, 45:611–639, 1997. doi: 10.1046/j.1365-2478.1997.440282.x.
- A. Ziolkowski, G. Parkes, L. Hatton, and T. Haugland. The signature of an air gun array: Computation from near-field measurements including interactions. *Geophysics*, 47(10):1413–1421, 1982. doi: 10.1190/1.1441289.

- 
- S. Zu, J. Cao, S. Qu, and Y. Chen. Iterative deblending for simultaneous source data using the deep neural network. *Geophysics*, 85(2):V131–V141, 2020. doi: 10.1190/geo2019-0319.1.







Graphic design: Communication Division, UIB / Print: Skjipes Kommunikasjon AS



[uib.no](http://uib.no)

ISBN: 9788230842805 (print)  
9788230840603 (PDF)