Focal deblending

Using the focal transform for simultaneous source separation

Kontakis, A.

DOI
10.4233/uuid:60f45315-7ad9-4188-b76e-cbe0b0af6c27

Publication date
2023

Document Version
Final published version

Citation (APA)
Kontakis, A. (2023). *Focal deblending: Using the focal transform for simultaneous source separation*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:60f45315-7ad9-4188-b76e-cbe0b0af6c27

Focal deblending

Using the focal transform for simultaneous source separation

Focal deblending

Using the focal transform for simultaneous source separation

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus
Prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op vrijdag 14 april 2023 om 12:30 uur

door

Apostolos KONTAKIS

Master of Science in Electrical Engineering
Technische Universiteit Delft
geboren te Athene, Griekenland

Dit proefschrift is goedgekeurd door de promotoren:
Prof. dr. ir. E.C. Slob en Dr. ir. D.J. Verschuur

Samenstelling promotiecommissie bestaat uit:

Rector Magnificus,	voorzitter
Prof. dr. ir. E.C. Slob,	Technische Universiteit Delft, promotor
Dr. ir. D.J. Verschuur,	Technische Universiteit Delft, promotor

Onafhankelijke leden:

Prof. dr. ir. G.J.T. Leus,	Technische Universiteit Delft
Prof. dr. W.A. Mulder,	Technische Universiteit Delft
Prof. dr. J. Schleicher,	UNICAMP, Brazilië
Dr. ir. G. Blacquière,	Technische Universiteit Delft
Dr. M. Ravasi,	KAUST, Saoedi-Arabië

ISBN 978-94-6366-677-0

The work in this dissertation was conducted at the Computational Imaging (CI) section, Faculty of Applied Sciences, Delft University of technology and was financially supported by the Delphi Consortium.

Typesetting system: LaTeX.

Printed in The Netherlands by Gildeprint.

To all the teachers I have ever had

Contents

Summary

Nearly-simultaneous-source (blended) acquisition differs from conventional acquisition in that seismic wavefields originating from different sources are allowed to overlap in the recorded seismic traces. This allows more flexibility in deciding the number of shots, the shot density and the effective acquisition time of a survey, but it adds the complication of having to handle blended wavefields.

This thesis explores an inversion-based deblending method for wavefield separation in the marine setting. As deblending is usually an underdetermined problem, extra information in the form of additional constraints and regularization is needed to obtain a unique solution with minimal blending-noise leakage. To this end, the proposed method uses the focal transform in combination with sparsity-promoting regularization to discriminate against solutions to the blending equation that are valid, but contain excessive amounts of blending noise. The focusing operation provided by the focal transform will tend to focus the coherent signal to be extracted but will not focus equally well incoherent blending noise. Sparse solutions will tend to retain the high-amplitude focused events but not the lower-amplitude blending noise. A key feature that makes sparse solutions possible is the ability to describe curved events in a subsurface-consistent manner, using few focal domain coefficients.

The focal transform can be defined in multiple ways, using one-way or two-way wavefield propagation operators. In the implementations described in this thesis, I use a crude velocity model, based on picked normal-moveout (NMO) stacking velocities, to construct focal operators that can focus surface data onto a set of depth levels where significant reflectors are found. This choice of velocity model is suboptimal for focusing purposes, but is a pragmatic compromise, given that a more detailed velocity model may not be available at the deblending phase of the processing workflow.

In principle the focusing and defocusing operations involve the entire dataset, which makes the focal transform computationally expensive to evaluate. An investigated remedy is to use acquisition-specific subsets of the input data to split the problem in smaller chunks, combined with a suitable flavor of the focal transform and focal grid. Another method extension that I discuss is that of using a focal-curvelet hybrid transform for deblending. The main advantage is that events with linear moveout tend to be more sparsely represented in a curvelet basis. However, this comes at the cost of extra computational effort and some difficulty in balancing the contribution of the two transforms to the final solution.

I test these approaches on both synthetic and field data, with examples on towed streamer and ocean-bottom-node acquisitions. While in most cases a perceptible amount of blending-noise leakage remains present in the results, a significant amount of blending noise is suppressed. In some cases the deblending process is able to uncover weak events previously masked by strong blending noise. When the hybrid transform is used, the results show a better recovery of events that are filtered out when the focal transform is used alone. Curved near offset events are in some cases also recovered with higher fidelity compared to using the curvelet transform alone.

A significant challenge is the sometimes limited focusing for field data and synthetics as a result of trying to approximate the kinematics of complex 3D velocity models with flat-layered models and stacking velocities. The computational cost of the method is also a challenge. While working with data and focal domain subsets helps, additional measures are needed before applying focal deblending on realistically-sized field data. I make several suggestions for modifications of the method and propose extensions for future research.

Samenvatting

Seismische acquisitie met bijna-simultane bronnen (of 'blended acquisition') verschilt van conventionele acquisitie in die zin dat seismische golfvelden afkomstig van verschillende bronnen elkaar mogen overlappen in de opgenomen seismische metingen. Dit biedt meer flexibiliteit bij het bepalen van het aantal opnamen, de opnamedichtheid en de effectieve acquisitietijd van een seismische campagne, maar het voegt de complicatie toe van het omgaan met overlappende ('blended') golfvelden.

Deze dissertatie onderzoekt een op inversie gebaseerde 'deblending'-methode voor golfveldscheiding in seismische data opgenomen op zee. Aangezien deblending gewoonlijk een ondergedetermineerd probleem is, is extra informatie in de vorm van additionele beperkingen en regularisatie nodig om een unieke oplossing te verkrijgen met een minimaal verlies van informatie. Daartoe gebruikt de voorgestelde methode de focale transformatie in combinatie met een spaarzaamheidsbevorderende regularisatie om oplossingen van de datavergelijking die geldig zijn, maar een te grote hoeveelheid mengruis bevatten, te discrimineren. De door de focale transformatie verschafte focussering zal ertoe leiden dat het te extraheren coherente signaal wordt gelokaliseerd, hetgeen niet geldt voor de incoherente mengruis. Spaarzame oplossingen zullen de gefocuste metingen met een hoge amplitude behouden, maar niet de mengruis met een lagere amplitude. Een belangrijk kenmerk dat spaarzame oplossingen mogelijk maakt, is de mogelijkheid om seismische aankomsten met sterke kromming te beschrijven op een ondergrond-consistente manier, met gebruikmaking van weinig coëfficiënten van het focale domein.

De focale transformatie kan op verschillende manieren worden gedefinieerd, met gebruikmaking van een- of tweerichtings- golfveldvoortplantingsoperatoren. In de in dit proefschrift beschreven implementaties gebruiken we een ruw snelheidsmodel, gebaseerd op gekozen zogenaamde 'normal-moveout' (NMO) snelheden, om focale operatoren te construeren die oppervlaktegegevens kunnen focussen op een reeks dieptes waar significante reflectoren worden verwacht. Deze keuze van het snelheidsmodel is suboptimaal voor focusseringsdoeleinden, maar is een pragmatisch compromis, aangezien een meer gedetailleerd snelheidsmodel mogelijk niet beschikbaar is in de deblendingfase van het seismische verwerkingsproces.

In principe hebben de focus- en defocusoperaties betrekking op de gehele dataset, waardoor de evaluatie van de focale transformatie computationeel duur wordt. Een onderzochte remedie is het gebruik van acquisitiespecifieke subsets van de input data om het probleem in kleinere brokken op te splitsen, gecombineerd met een geschikte variatie van de focale transformatie en het gekozen focale raster. Een andere methode-uitbreiding die besproken wordt is het gebruik van een focale-curvelet hybride transformatie voor deblending. Het belangrijkste voordeel is dat seismische reflecties met lineaire tijd-afstand aankomsten doorgaans spaarzamer worden weergegeven in een curveletbasis. Dit gaat echter ten koste van extra rekenwerk en enige moeite om de bijdrage van de twee transformaties aan de uiteindelijke oplossing in evenwicht te brengen.

Deze benaderingen worden op zowel synthetische als veldmetingen getest, met voorbeelden van opnamen via gesleepte streamers en via oceaanbodem-sensors. Hoewel in de meeste gevallen een merkbare hoeveelheid mengruis in de resultaten aanwezig blijft, wordt een aanzienlijke hoeveelheid mengruis onderdrukt. In sommige gevallen kan het deblendingproces zwakke seismische reflecties reconstrueren die in de ruwe data door sterke mengruis werden gemaskeerd. Wanneer de hybride transformatie wordt gebruikt, laten de resultaten een beter herstel zien van seismische reflecties die worden weg gefilterd wanneer de focale transformatie alleen wordt gebruikt. Gekromde seismische reflecties bij de kleine bron-ontvanger afstanden worden in sommige gevallen hersteld door de hybride transformatie met een hogere nauwkeurigheid dan wanneer alleen de curvelet-transformatie wordt gebruikt.

Een belangrijke uitdaging is de soms beperkte focus voor gesimuleerde en veldmetingen als gevolg van pogingen om de kinematica van complexe 3D-snelheidsmodellen te benaderen met vlakke gelaagde modellen en NMO-snelheden. Ook de rekenkosten van de methode vormen een uitdaging. Hoewel het werken met slimme subsets van de data en van focale domeinen helpt, zullen aanvullende maatregelen nodig zijn alvorens focale deblending toe te passen op veldmetingen van realistische omvang. Ik doe verschillende suggesties voor wijzigingen van de methode en stel uitbreidingen voor toekomstig onderzoek voor.

1

Introduction

1.1 The seismic experiment

A number of human activities present to us the challenging problem of measuring material properties of the subsurface. They sometimes take the form of prospecting, e.g. for mining sites, hydrocarbon reservoirs, suitable sites for geothermal energy extraction and near-surface investigation for construction purposes, for example for windfarm installations. In other cases the objective is to monitor an ongoing process, such as changes in hydrocarbon reservoirs, absorption of carbon dioxide in underground-sequestration efforts and hydrogen storage. What these applications have in common is that they require information about the subsurface over an area. Although it may be technically feasible to drill and recover samples for direct inspection, drilling can be very expensive and is generally used very sparingly. It provides us with high quality information, yet that information is very localized at the drilling location. On its own, drilling is often not capable of giving us the big picture.

Fortunately, we can infer lots of information about the subsurface indirectly. For example, gravimetry [Toushmalani and Saibi, 2015] and gravitational gradiometry [Pawlowski, 1998] measurements can be used to extract information about variations in the Earth's density, by measuring the Earth's gravitational field or variations in gravitational acceleration, respectively. Controlled-source electromagnetic (CSEM) and transient electromagnetic (TEM) methods can be used to infer an electric resistivity model of the subsurface [Anderson et al., 2008]. Seismic methods are popular when it comes to hydrocarbon prospecting but are also used for other applications such as assessing the shallow subsurface for geotechnical projects and monitoring carbon capture. Seismic methods rely on elastic wave propagation to recover information such as compressional/shear wave velocities and the location of reflecting layer boundaries [Telford et al., 1990]. Note that, generally, the quantities we would like to know (e.g. coordinates of water-oil contacts) are not usually the ones we actually measure (e.g. pressure and particle velocities): the former have to be inferred from the latter, using a mathematical model that describes their relationship. Often this is done in multiple stages, using a chain of models and intermediate model parameters. Therefore, the quality of our inferred measurements will depend on the quality of each one of the mathematical models in the chain, in addition to the quality of what we directly measure. This is a price we have to pay for the convenience of measuring by proxy.

For the rest of this thesis we will turn our attention to reflection seismology using active sources, one of the multiple ways of using mechanical waves for subsurface exploration. To conduct a reflection seismology survey over an area of interest, two types of devices are needed, at the very least: a seismic source for producing mechanical waves and a seismic receiver[1] for detecting them. In active surveys the source is human-controlled, in contrast to passive sources, where we rely on natural seismic activity to act as a source.

Sources will typically be vibrating plates coupled to the ground, for land acquisition, or airguns, for marine acquisition. The receivers usually take the form of geophones or hydrophones. Geophones measure one or more vector components of particle velocity[2]. Geophones will be used for land acquisition and for some types of marine acquisition. Hydrophones measure pressure and are used in marine acquisition. An overview of different types of sources and receivers can be found in Telford et al. [1990, chapter 4.5].

The basic idea behind reflection seismology, as the name suggests, is to exploit the natural phenomenon of wave reflection. A seismic source is used to generate a mechanical wave that propagates towards the interior of the Earth. At the boundaries, where parts of the subsurface with different acoustic impedance meet, part of the propagating wavefield will reflect and part of it will refract into the next layer. Eventually, after a series of refraction and reflection steps, a portion of the wavefield will travel toward the surface, where it will be recorded by the receivers. The process is repeated after moving the sources and/or receivers to different locations [Telford et al., 1990]. A schematic depiction of the process can be seen in figure 1.1.

The recorded seismic data can provide us with a host of information about the subsurface. For example, seismic migration can be used to map the wavefield recorded at the surface to a set of reflectors in the subsurface. These reflectors will be located at the boundaries of structures with different material properties. Using migration, we can therefore extract structural knowledge about the interior of the earth. Seismic migration has been studied extensively and is still a very active topic of research.

---

[1] The term 'detector' is often used in literature instead of 'receiver'.

[2] There exist geophones based on MEMS technology that measure particle acceleration instead of particle velocity [Wei, 2013].

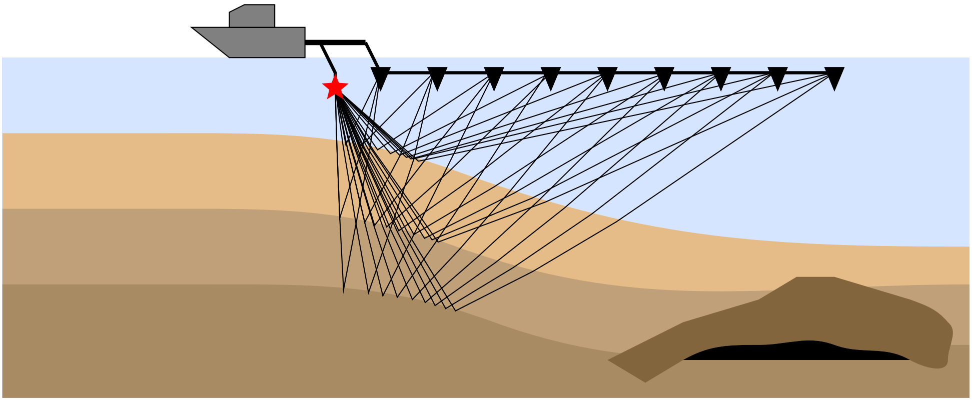


Figure 1.1: *Schematic depiction of a seismic experiment in reflection seismology in a marine environment. Red stars and black triangles represent sources and receivers, respectively. Only primary reflections are depicted in this example.*

As a result, there exist many methods for migrating seismic data, ranging from the very simple to the very sophisticated. The reader may refer to Bednar [2005]; Biondi [2006]; Robein [2010] for more information on seismic migration methods. Note that an important prerequisite step for migration is the estimation of the wave propagation velocities of the various subsurface layers, using methods like full waveform inversion (FWI) [Gauthier et al., 1986; Virieux and Operto, 2009], migration velocity analysis (MVA) [Stork and Clayton, 1991], or tomography [Farra and Madariaga, 1988; Zhang and Toksöz, 1998; Billette and Lambaré, 1998]

The journey rarely ends at reflectivity and velocity, which are *wave*-related properties. In the context of hydrocarbon exploration and production, carbon dioxide injection and geothermal exploration what we are usually interested in is *rock*-related properties, such as porosity and fluid saturation. There exist several models that map rock properties to wave-related properties [Mavko et al., 2009], making it possible to get petrophysical information out of seismic data. The presence of hydrocarbons in an area can be then assessed, as well as good candidate drilling locations. Seismic surveys have also applications in already producing fields, were they are used to monitor changes at the reservoir level [Lumley, 2001].

1.2 Conventional vs blended acquisition

When a seismic acquisition is taking place, a series of shots is fired in succession, generating wavefields that get recorded by the receivers. The receivers may start recording whenever a source is activated, or they may be recording continuously, depending on the acquisition type. An important decision made at the planning phase is at which time instant each of the shots should be fired. Or, equivalently, how much time one must wait between each two successive shots, a quantity I will refer to as shot time interval. The answer will usually depend on other parameters of the acquisition. For example, in a marine streamer survey the shot time interval will be determined by the boat speed, the airgun recharge time and the desired shot density. For land acquisitions that use a vibratory source, the sweep time will place a lower bound on the shot time interval, etc.

Whenever a shot is fired, a propagating wavefield is generated at the shot location. In conventional acquisition, the shot time interval is chosen to be large enough, such that when recording the wavefield from one shot, there will be at most minimal crosstalk (i.e. below noise level) from the wavefields generated by the preceding and following shots, for the duration of the shot record. Essentially, the objective is to record the wavefield due to each shot separately from those due to the other shots. A seismic acquisition adhering to this non-interference condition is called an *unblended* acquisition. An acquisition where this condition is violated, i.e. where the receivers record a superposition of wavefields generated by different shots, is referred to as *blended* acquisition [Berkhout, 2008], also known as simultaneous, or near-simultaneous, acquisition [Beasley, 2008; Howe et al., 2008]. Its purpose is to increase the efficiency of acquisition without sacrificing too much accuracy in the final processing results [Abma and Foster, 2020].

Planning a blended acquisition requires making a few decisions regarding the sources. More specifically, the location, activation time, and signature of the source must be decided for each shot. The combination of these parameters defines the *blending code* used for the acquisition. Knowledge of the blending code is almost always a prerequisite for further processing blended data. This is especially true when the overlapping wavefields are to be explicitly separated before further processing, a process usually referred to as *deblending*, or source separation [Abma and Foster, 2020]. I will return to the important subjects of the blending code and deblending in chapter 2.

Figure 1.2 shows a small example of data from an unblended and a blended acquisition. This is an example using synthetic data, generated using a 2D slice from a modified version of the SEG/EAGE overthrust velocity model [Fred Aminzadeh, 1997]. In both kinds of acquisition the objective is to record the wavefield generated by a shot at $x = 4$ km and a second shot at $x = 6$ km, for a duration of approximately 2 s. The wavefield is recorded at the surface of the model, by receivers forming a total aperture of 3 km. Figure 1.2a displays the unblended data. Each column of pixels denotes a seismic trace recorded by a receiver in time, the pixel intensity corresponding to a pressure measurement. Because a sufficiently large shot time interval was used, it is possible to separately record the wavefield created by each of the sources. The situation changes, however, in figure 1.2b. Here the shot time interval was 0.3 s, which is not enough for recording the two wavefields independently. This

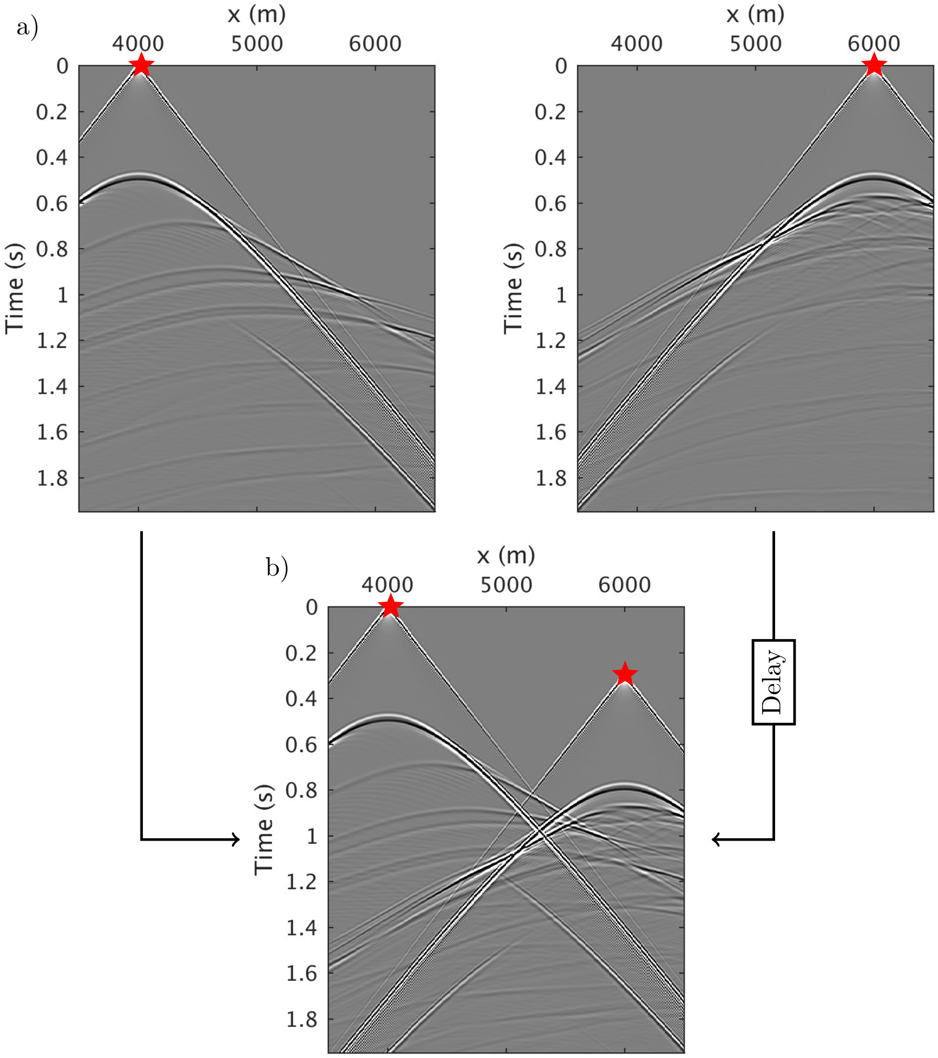


Figure 1.2: *a) Two unblended shot gathers recorded using conventional acquisition; b) The same shot gathers as one blended gather.*

makes the data blended, and the acquisition a blended acquisition.

A snapshot of the total wavefield inside the medium at $t = 1.0$ s can be seen in figure 1.3. Here the wavefield is frozen in time, its values in space shown superimposed on the velocity model. Figure 1.3a shows the wavefield as it is when only one source is activated, as is the case in conventional acquisition. In figure 1.3b a second source is activated shortly after the first, leading to a superposition of two wavefields. Note that in actual acquisition, unlike the synthetic case shown here, we do not have access to the pressure values everywhere in the subsurface, but only on a limited number of locations where receivers are placed. Another complication that is to be found in real data but is absent from the example in figure 1.3 is the presence of surface-related multiples. In realistic processing scenarios these cannot be removed prior to deblending.

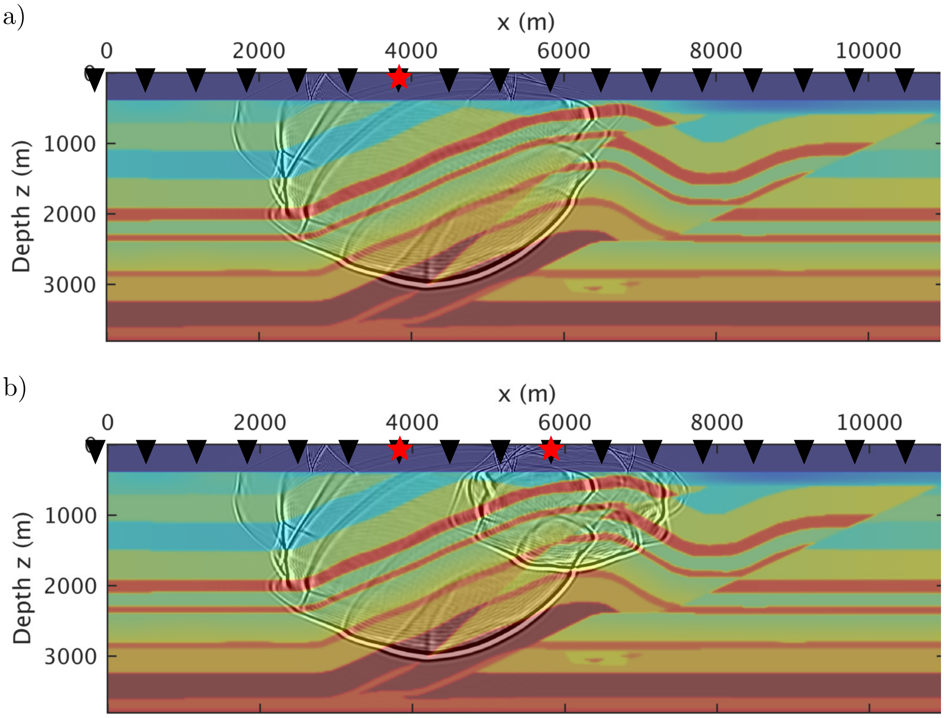


Figure 1.3: *Total wavefield at* $t = 1$ *s when using a) conventional and b) blended acquisition.*

Because crosstalk is kept to a minimum, unblended acquisition will yield data that present fewer risks, such as crosstalk noise masking important weak events. For this

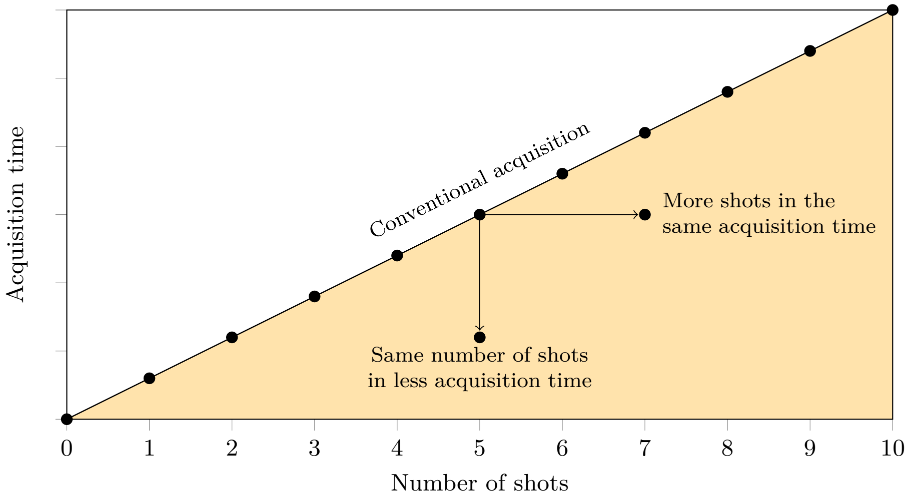


Figure 1.4: *Number of shots vs acquisition time for conventional and blended acquisition. Any point within the orange-colored area represents a blended acquisition with a different tradeoff between acquisition time and number of shots.*

reason it is still the norm. It has, however, an important drawback: the relationship between the number of shots and acquisition time is inflexible. There is a maximum number of shots that can be fired in a given amount of time, determined by the shot time interval. It is not possible to exceed this number, because that would necessarily introduce crosstalk. Avoiding crosstalk often comes at the price of reduced shot density. Beasley [2008] gives an example of a complex marine acquisition scenario, where multiple vessels are used. To avoid crosstalk, only one vessel can fire its source at any time, while the others wait. This implies a coarser shot spatial interval, which leads to aliasing problems that need to be treated in processing, or to a longer survey duration due to lower vessel speed, which makes the survey more expensive.

By allowing crosstalk, blended acquisition allows for a more decoupled relationship between number of shots and total acquisition time. Figure 1.4 demonstrates this relationship. The slanted line represents conventional unblended acquisition, its slope determined by the shot time interval. The orange area contains the combinations of acquisition time and number of shots that become possible in a blended acquisition.

Blended acquisition offers several advantages over conventional acquisition:

- More flexibility in deciding the number of shots, the shot density and the effective acquisition time of a survey [Abma and Foster, 2020]. Multiple acquisition geometries can benefit from this. Using a synthetic example, Gulati et al. [2011] show that blending is a viable option for getting vertical seismic profile (VSP) data faster. Beasley et al. [2012] test a blended variant of the narrow azimuth (NAZ) geometry, with positive initial results. Moldoveanu et al. [2012] demonstrate the viability of blended multivessel coil shooting, using synthetics. Higher efficiency is also reported in Walker et al. [2013] for the ocean bottom seismic (OBS) geometry. Long et al. [2013] propose the simultaneous long offset (SLO) acquisition, a blended form of the continuous long offset (CLO) geometry that decreases the spatial interval between two shot locations [van Mastrigt et al., 2002].

- Reduced cost per trace, due to the reduction in acquisition time for the same number of traces. Note that although reduced acquisition time may lead to savings in certain areas such as labor cost and fuel consumption, it is not always a given that the total cost of the acquisition is reduced significantly, see e.g. Krupovnickas et al. [2012].

- Increased signal-to-background-noise ratio [Moore et al., 2013; Berkhout and Blacquière, 2013]: This happens because the amount of background noise recorded depends only on the effective survey time and not on the blending process. If the number of shots is kept constant, a blended survey takes less time than an unblended one, so the amount of background noise is reduced. If the acquisition time is kept constant, a blended survey will record more shots. Then, the amount of background noise would be the same for both conventional and blended surveys, but the blended one will contain more signal. This makes blended acquisition beneficial for areas with high background noise.

- Better illumination of the subsurface [Berkhout et al., 2010; Abma and Foster, 2020], owing to denser sampling in the source dimension and/or wider aperture. This leads to a more complete angle coverage of subsurface grid points [Blacquière et al., 2012].

- Easier survey scheduling and compliance with environmental regulations. Scheduling a lengthy survey can be difficult in crowded areas, where exploration efforts are very active. Another example is exploration in the Arctic region, where the long winters and short summers may place additional constraints on survey scheduling. Reduced acquisition time means that a suitable time window for conducting the survey is more likely to be found.

The popularity of blended acquisition for field surveys differs for the land and marine cases. Its adoption for land surveys happened much earlier than it did for the marine case. There are two main reasons for this, that have to do with seismic source technology and the ease of adding more sources in the acquisition. Vibratory sources, such as those used for land acquisition, are much more flexible in the kind of source signatures they can generate, something that can be used to the advantage of blended acquisition [Howe et al., 2008]. Impulsive sources, on the other hand, are much more limited in this aspect [Beasley et al., 2012]. Also, extra sources are cheaper and easier to accommodate in a land survey than in a marine one. Despite this fact, the advantages of blended acquisition have prompted a number of marine field trials [Abma et al., 2012; Zhang et al., 2013a; Makhorin et al., 2013; Henin et al., 2015; Walker et al., 2017; Crosby et al., 2022].

Perhaps the ultimate test for blended acquisition is whether the acquired data are suitable for reservoir characterization and monitoring applications. Reservoir characterization requires that amplitudes preserve amplitude-vs-offset/angle (AVO/AVA) effects. Also, the very weak reflection events originating from the reservoir level must be as noise-free as possible. There have been some encouraging results on that front. An AVO analysis on blended OBC data from Trinidad showed good agreement with predictions based on well data [Paramo et al., 2013]. Li et al. [2019] compared processing results from a conventional and a blended marine survey carried out over the same area. After deblending, Li et al. [2019] show that the AVO responses from the blended survey are nearly identical to those from the conventional one.

For reservoir monitoring/timelapse acquisition, amplitude fidelity is even more important, as we are interested in recovering changes at the reservoir level that might have a very minute imprint on the recorded seismics. Naturally, one would be very sceptical about using a blended survey for such applications, for fear of excessive contamination of the signal with blending noise. Interestingly, initial feasibility studies on reservoir characterization show positive results. In Krupovnickas et al. [2012], the authors mention positive results for shallower horizons, but extracting a 4D signal from deeper horizons was challenging with their processing setup. Shipilova et al. [2016] also report positive timelapse results from blended data.

At the time of writing, both blended and conventional seismic acquisition is generally done using multiple identical broadband sources. Owing to advances in seismic source technology, it is possible that in the future broadband sources will be replaced by multiple narrowband sources, each one optimized for a different part of the spectrum. This concept, referred to as dispersed source arrays (DSA), can be combined with blended acquisition. The result is a new form of blending [Berkhout, 2012], where the blending code is extended to specify individual frequency bands. The combination of DSA and blended acquisition paves the way for automated, decentralized acquisition [Berkhout, 2013]. DSA field trials and associated data processing are discussed in Tsingas et al. [2016, 2020]; Jeong et al. [2022].

1.3 Deblending in the literature

The advantages of blended acquisition come at the cost of additional noise, in the form of crosstalk, that must be dealt with in the seismic data processing chain. I will refer to this crosstalk as *blending noise*, to differentiate it from other sources of noise present in the seismic signal. There are, generally speaking, three distinct approaches in how blending noise can be handled. The first approach is to deblend the data, i.e. to separate the overlapping wavefields such that each constituent wavefield is associated with one particular source. The second approach is to simply disregard the presence of blending noise and use the blended data as direct input for other types of processing, e.g. for migration. The third approach is applicable to inversion-based processing in which the seismic data is present in the misfit term of the objective function. An example is full waveform inversion. In such cases it is possible to incorporate blending within the objective function and use the blended data directly as input.

In many cases, especially for marine blended data, the blending code on its own cannot provide sufficient suppression of blending noise. In those cases, deblending may need to be applied as an extra processing step. Many deblending algorithms have been proposed in literature for this purpose, and they can be generally divided in two categories. Algorithms of the first category rely on denoising tools to remove blending noise. Algorithms of the second category pose deblending as a constrained inversion problem. Methods from each of these two categories are increasingly incorporating deep learning techniques to achieve better wavefield separation.

■ 1.3.1 Blending codes

The blending code itself can be turned into a very effective tool for preprocessing blended data. Much research has been devoted to studying the performance of existing source coding schemes, as well as to developing new ones. The main goal of a blending code is to define groups of shots fired (near) simultaneously, and assign a code to each of the sources in the group that is as orthogonal as possible to the codes of the other sources in the group. Then, the wavefield associated with each source is partially distinguished from the rest of the wavefields, by using the fact that it was coded differently. This can already provide a first level of separation. As a concept, it has similarities with a code division multiple access (CDMA) scheme, as used in cellular telecommunications, where concurrent transmissions by different users are also separated by means of orthogonal codes [Rao and Dianat, 2005, Chapter 3].

As mentioned earlier, vibratory sources are more flexible when it comes to the generated wavelet, i.e. the waveform they generate. Consequently, coding schemes tailored to them have been studied for a longer time than schemes for impulsive sources. Codes proposed in the literature use techniques such as complementary series with sweep polarity switching [Silverman, 1979; Garotta, 1983; Pritchett, 1991], sweeps with different phase offsets [Landrum, 1987; Sallas et al., 1998], modified Gold codes [Sallas et al., 2008; Dean, 2012] and m-sequences [Wong, 2013; Wong and Langton, 2014, 2015]. Although these approaches are used in land acquisition, there is an ongoing effort to produce better vibratory sources for the marine environment [Pramik et al., 2015; Schostak and Jenkerson, 2015; Dellinger et al., 2016]. This means that these, or variations of these techniques may become practical for marine surveys in the future.

Although blending codes for impulsive sources have not been researched as intensely as those for vibratory sources, this is changing, as marine blended acquisitions become more popular. A first type of code exploited the spatial distance between the shots, to facilitate separation via dip discrimination [Beasley, 2008]. Another approach assigns time delays to different sources. These time delays are most often (but not always) random [De Kok and Gillespie, 2002]. An advantage of random time delays is that blending noise has then an impulsive and incoherent character and can be treated with denoising tools. A special form of random time delay is to add jittering to regular shot times [Moore et al., 2008], which can be combined with spatial jittering of the shot location [Wason and Herrmann, 2013]. In an effort to make codes for impulsive sources more orthogonal, more shots can be fired per location [Mueller et al., 2015; Wu et al., 2015; Mueller et al., 2016].

Time delays can also be deterministic; Poole et al. [2014] describe a firing pattern that uses two sources that fire one after the other, with a constant time interval between their respective activation. The order in which the sources fire alternates after each shot. The authors propose a deblending algorithm that takes advantage of this firing pattern. Robertsson et al. [2016a,b] propose a different pattern, named 'seismic apparition', that employs constant time delays. There, the second source alternates every shot between firing simultaneously with the first source, or after a constant time interval. Using this blending code, perfect separation is possible in the frequency-wavenumber (FK) domain, up to a certain frequency determined by the shot spacing. Zu et al. [2016b] investigate periodically varying delays and show an example where they lead to a better deblending result than random time delays.

■ 1.3.2 Denoising methods

A crucial function of the blending code is to make blending noise incoherent. This is usually done by sorting in a domain other than common shot, e.g. common receiver, common offset or common midpoint. Since the signal, in general, will be coherent, the blending code can help us differentiate between signal and noise, as long as we have a way to filter out the incoherent part (figure 1.5). After sorting, the blending noise will have an impulsive character in space, under the effect of the blending code. It is possible then to do deblending by leveraging traditional tools used for impulsive noise removal, such as median filters. Median filters are non-linear filters that are robust against data outliers. Careful application of median filtering can suppress impulsive blending noise, which often behaves as an outlier within a small data window, by having significantly different amplitude than the neighboring samples [Liu et al., 2009; Huo et al., 2012; Zhang et al., 2013b; Chen, 2014; Liu et al., 2014; Gan et al., 2015; Zhan et al., 2015]. [Baardman and van Borselen, 2012; Chen et al., 2015] apply a median filter after normal moveout (NMO) correction. By removing curvature from seismic events, NMO correction can protect parts of the

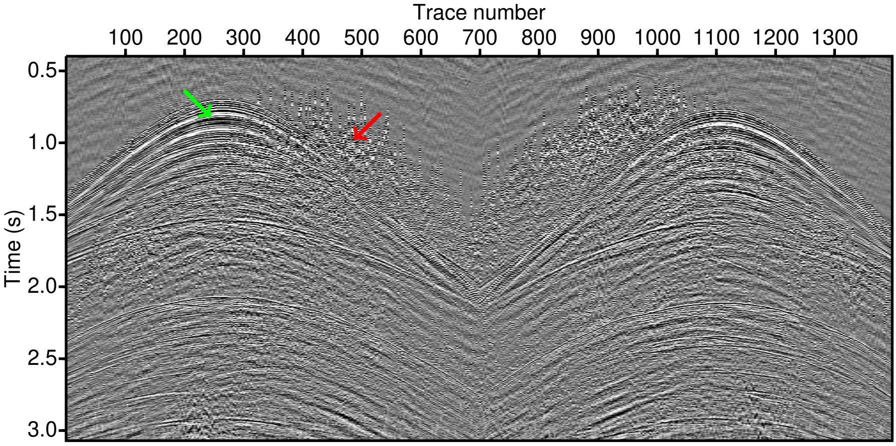


Figure 1.5: *Example of a common receiver gather of pseudo-deblended data from an OBN dataset. The signal to be extracted is coherent (green arrow), appearing as events that have continuity. Due to the blending code, blending noise (red arrow) is not coherent and is sometimes treated using methods designed to remove high-amplitude outlier samples from seismic traces.*

signal that may erroneously be removed as noise by the median filter. Chen et al. [2015]; Peng et al. [2013] use an additional FK filter as a coherency pass filter, for further noise suppression. [Yu et al., 2017] combine NMO correction with filtering in the wavelet domain for blending noise suppression. Spitz et al. [2008] use prediction error filters to remove blending noise, drawing inspiration from methods used for removing seismic events related to multiple scattering. Andersson et al. [2016] pair a seismic apparition blending code with a dealiasing algorithm to deblend. [Sun et al., 2020, 2022] trained convolutional neural networks (CNN) to map data contaminated with blending noise to clean, denoised data.

■ 1.3.3 Inversion-based methods

A different approach to deblending is to pose it as an inversion problem. Deblending then becomes a problem of finding a dataset that, when blended with the same code as the one used in the field, the result matches the recorded blended data. The inversion problem is *ill-posed*: there exist an infinite number of solutions that will match the recorded blended data after applying the blending code to them. The inversion must be constrained, such that the recovered solutions have minimal blending noise leakage as well as minimal signal harm.

Wapenaar et al. [2012] propose a separation method that involves calculating a modified, band-limited pseudo-inverse of the blending operator. This is possible when the source dimension is sampled densely enough and the group of sources participating in the blended experiment are spatially close. The most common approach is to base the separation on some form of coherency-based noise suppression during iterative inversion. For this purpose, Ayeni et al. [2009] use non-stationary dip filters. Doulgeris et al. [2010]; Mahdad et al. [2011]; Doulgeris et al. [2012b] use a flexible combination of thresholding and FK coherency pass filters to progressively estimate and remove blending noise. Zu et al. [2017] use plane wave destruction filtering [Fomel, 2002] in a combination with least-squares inversion to achieve separation. Leader and Biondi [2014] extend the imaging condition of reverse time migration (RTM) and use RTM as a deblending tool. Beasley et al. [2016]; Moore et al. [2016, 2017] suggest replacing the inner product operation with a robust version that discriminates against outliers such as impulsive blending noise, guiding the inversion process to a more favorable solution.

The deblended solution is expected to represent a sampled seismic wavefield and must, therefore, have the structure of one. This is extra information that can be used to constrain the solution, as long as there is a way to mathematically describe this structure. Perhaps the most intuitive way to do so is to express the deblended solution as a linear combination of simpler structures, such as plane waves, that act as building blocks. Borrowing terms from compressive sensing, we will refer to these building blocks as *atoms*, that collectively form a *dictionary*. Ideally, the dictionary should be selected such that most of the signal is expressed as a linear combination of few atoms, i.e. such that the deblended solution has a *sparse* representation in that dictionary. By placing sparsity constraints or using sparsity-promoting regularization in the optimization, a solution with minimal noise leakage is potentially reached, under the assumption that a solution containing noise is less sparse than a noise-free one.

Perhaps the most crucial choice that needs to be made for such a sparsity-based method is that of the dictionary. Usually the dictionary is chosen to be the basis or frame of a mathematical transform, with all the attractive properties that this entails, such as computationally fast implementations and quick convergence rates. Transforms used for sparsity-based deblending include Fourier [Sen et al., 2014; Abma et al., 2015], seislets [Chen et al., 2014; Chen, 2015; Gan et al., 2015], curvelets [Zu et al., 2016a], the Radon family [Akerberg et al., 2008; Ayeni et al., 2011; Trad et al., 2012; Ibrahim and Sacchi, 2013, 2015] and multifrequency array steering [Ji et al., 2012]. Transform combinations are also possible; Peng and Meng [2016] use both a linear Radon and a wavelet transform in their deblending process. Instead on basing their dictionary on mathematical transforms, Zhou et al. [2013] use dictionary learning to extract it from seismic data.

Another way to attack the ill-posed nature of the deblending problem is to assume that the deblended solution will be *low-rank*, i.e. few singular values will have significant magnitude when calculating a singular value decomposition. This can be viewed as an alternative form of sparsity that operates on singular values. This form of sparsity provides us an extra avenue for constraining the deblending problem. An attractive feature of rank reduction methods is that they do not require choosing a dictionary. They do, however, require sorting the elements of the solution into a matrix (or higher-order array) that exhibits the low-rank property. A popular choice is to form a Hankel matrix out of monochromatic frequency slices of data in the frequency-space domain [Maraschini et al., 2012; Cheng and Sacchi, 2013, 2015; Chen et al., 2016]. Another approach is to sort the data based on midpoint/half offset coordinates and construct a hierarchically semi-separable matrix from frequency slices [Wason et al., 2014; Kumar et al., 2015]. Properly resorted time domain slices are also an option, as demonstrated by Kumar et al. [2016]. Since calculating large singular value decompositions is expensive, computationally cheaper alternatives are sometimes preferred. For this reason, Cheng and Sacchi [2016] replace singular value decompositions with randomized QR decompositions.

Iterative, inversion-based deblending is sometimes combined with deep learning solutions. Zu et al. [2020] use an iterative inversion similar to that of Mahdad et al. [2011], but with deblended data estimates generated by a combination of convolutional and deconvolutional neural network layers. Wang et al. [2022a] use iterative inversion and a multi-resolution U-Net to take advantage of the multiscale nature of seismic data. Neural networks can take the form of deep preconditioners that can be used to steer the deblending inversion [Xu et al., 2022]. A typical requirement of neural network-based methods is the availability of training data. For deblending in particular, this means that either unblended or adequately deblended data must be available for training. Luiken et al. [2022] propose a method that obviates this requirement by combining the alternating method of multipliers (ADMM) with a self-supervised denoiser. Wang et al. [2022b] repurpose the blind-spot training technique, initially devised for image denoising, to perform deblending within an iterative inversion scheme.

■ 1.3.4 Using blended data directly

Instead of deblending the data as an explicit process, one can choose the alternative route of using the blended data as an input to processing algorithms[3] [Berkhout, 2008; Hampson et al., 2008; Verschuur and Berkhout, 2009; Berkhout et al., 2009]. When it comes to migration, the extra flexibility in the acquisition comes at a price of extra noise in the migration result. Jiang and Abma [2010]; Schuster et al. [2010] investigated the amount of crosstalk noise added to the image due to blending. [da Silva et al., 2012] studied the effect of applying different imaging conditions. A number of schemes are able to mitigate this type of noise. Tang and Biondi [2009] apply an ℓ_2-norm regularization on the inversion process. Dai et al. [2011] use a

---

[3] Pseudo-deblending will be usually required as a first step in the workflow in this case.

deblurring filter as a preconditioner in least-squares migration. Ferner and Sacchi [2014]; Castellanos et al. [2016] use M-estimators and a robust version of the imaging condition respectively, leveraging ideas from robust statistics. Li and Zhao [2015]; Li et al. [2016]; Chen et al. [2017] demonstrate the application of rank reduction techniques for crosstalk mitigation in RTM.

A technique sometimes employed for speeding up the migration of pre-stack unblended data is to numerically blend it beforehand and carry out the migration using the blended result. From the point of view of the migration algorithm, whether the data is blended in the acquisition phase or numerically on the computer does not make a difference. There are, however, some practical differences with respect to the blending code used in each case, as well as the behavior of random noise present in the records. When blending is done numerically, there are very few restrictions regarding the blending code to be used, which means that one can use highly orthogonal codes. For more information on the kinds of codes that are possible, the reader can refer to Godwin and Sava [2010, 2011]; Hu et al. [2011]; Schmidt et al. [2013]; Schleicher et al. [2013]. As we saw earlier, when blending happens at the acquisition phase, the choice of blending code is limited by the seismic source hardware and other acquisition-related restrictions. When it comes to random noise, there is an advantage in doing blending in the field rather than numerically, as demonstrated by Berkhout and Blacquière [2013].

FWI is sometimes sped up through similar numerical blending techniques [Ben-Hadj-Ali et al., 2009; Krebs et al., 2009; Boonyasiriwat and Schuster, 2010; Gao et al., 2010; Li et al., 2012]. As in migration, using the blended wavefield for FWI introduces artifacts in the result [Lee et al., 2012], which can be partially avoided by using a more robust objective function [Son et al., 2012]. The same technique can be also applied to speed up the generation of synthetic seismic data [Herrmann et al., 2009; Neelamani et al., 2010].

Two other processing steps that are commonly applied to marine seismic data prior to migration are surface-related multiple removal and source deghosting. The purpose of surface-related multiple removal is to remove reflection events that have reflected at least one time at the sea-air interface. This step is needed for suppressing imaging artifacts when the migration algorithm expects primary reflection-only datasets as an input. Doulgeris et al. [2012a] propose integrating deblending and multiple removal, the added benefit being that multiples do not have to be removed in an additional step. The objective of source deghosting is to remove the 'ghost effect', i.e. additional events that are generated by a 'ghost source' that manifests from the reflection of the direct wave at the sea-air interface. It can be thought of as a natural kind of blending [Berkhout and Blacquière, 2014] and, therefore, a properly modified deblending algorithm can handle deghosting as well, showing notch infill [Wu et al., 2016]. An alternative is to migrate blended data without prior removal of the surface-related multiples. Verschuur and Berkhout [2011] show that better illumination of the subsurface can be achieved in this way, and it removes the need for an extra surface-related multiple elimination step, making blending-aware a preexisting approach to seismic imaging [Whitmore et al., 2010].

1.4 Thesis objective

The main objective of this thesis is to study the feasibility of deblending marine seismic data using an inversion-type algorithm based on the focal transform. Moreover, I investigate two variants of the method: a hybrid focal-curvelet extension, as well as a modification to the focal transform aimed at reducing its computational cost.

The focal transform uses an approximate velocity model of the subsurface and wavefield extrapolation to transform a dataset recorded at the surface into a collection of datasets focused at chosen depth levels of the velocity model. The building blocks of the focal transform are events with hyperbolic moveout, translated in space and time. Due to the hyperbolic moveout they exhibit curvature near the apex, unlike other transforms that use plane waves, curvelets or seislet atoms. There are three main reasons that make the focal transform potentially attractive for deblending:

- The different behavior of signal and blending noise under the focusing operation. This property is used to discriminate signal from blending noise, achieving a deblended result. Sparsity constraints are be a key component to this.
- It enables us to bring prior information about the subsurface into the problem in order to constrain it.
- Kutscha and Verschuur [2012] demonstrated its effectiveness on the related problem of seismic data interpolation.

Focal deblending has some similarities with other deblending methods proposed in the literature. The apex-shifted hyperbolic Radon transform used by Ibrahim and Sacchi [2013, 2015] is similar to the single-sided focal transform acting on common receiver gathers. The two transforms are similar in that they both incorporate hyperbolic moveouts, the difference being that the focal transform uses wavefield extrapolators. In principle both methods could be modified to also handle non-hyperbolic moveouts. A significant difference exists when comparing with the double-sided focal transform, which unlike the hyperbolic Radon transform, uses both the source and receiver dimensions of the data simultaneously.

Leader and Biondi [2014] propose using a closed-loop process based on migration and demigration steps for deblending. The focusing and defocusing operations of focal deblending are conceptually very similar. The main difference between the two methods is that focal deblending is based on downward continuation and simple NMO velocity models. Leader and Biondi [2014] instead use RTM and more accurate velocity models. A similarity is that Leader and Biondi [2014] use an extended image domain, which is necessary for handling velocity inaccuracies. This is analogous to the local-offset dimension of focal subdomains.

1.5 Thesis outline

The thesis is organized as follows:

- Chapter 2: *Formulation of blending and deblending.* I first define the data cube structure and several operations associated with it. The data cube is used to sort 5D seismic data samples into a 3D array. I then describe the blending operator, using a formulation that is suitable for implementing many kinds of blending codes that are realistic for field acquisition. I examine more closely the specific case of random time delay codes. The adjoint operation, pseudo-deblending, is also introduced. Finally, I discuss a generic formulation of deblending as an optimization problem.

- Chapter 3: *Focal deblending.* The two variants of the focal transform are introduced. Although the aim of both transforms is to focus surface seismic data, they do so in a different manner. A comparison of the two transforms follows: each of the two transforms is suitable for different acquisition geometries, which will be important for chapter 4. I then present the effect of focusing on blending noise. As the extrapolation operators used for focusing act as FK filters, there are restrictions on what kind of events each operator can explain. I study this effect, with further discussion about it in chapter 5. Finally, I discuss the link between sparsity in the transform domain and blending noise leakage is.

- Chapter 4: *Dealing with 3D acquisition using smart subsets.* Seismic wavefields are rarely well-sampled in all dimensions. Even with the inevitable subsampling that stems from the requirements of practical acquisition, the total blended dataset might be quite large. As a way to partially overcome sampling and computational cost issues, this chapter introduces the idea of deblending suitable subsets of the dataset independently. I redefine focal deblending using subsets of the multi-streamer and the OBN acquisition geometries.

- Chapter 5: *Hybrid deblending solutions.* In this chapter I use the focal and curvelet transforms jointly for deblending. I discuss the advantages and disadvantages of using this hybrid transform and I present comparisons on synthetic data.

- Chapter 6: *Field data examples.* This chapter shows a number of deblending examples, using numerically blended field data. The examples are related to the deblending methods proposed in the previous chapters.

- Chapter 7: *Conclusions and discussion.* In this final chapter I present key conclusions followed by proposals for further research.
- Appendix A: *Focal deblending as a BPDN problem.* In this appendix I recast the focal deblending formulation as a canonical basis pursuit denoising (BPDN) problem, as defined by [van den Berg and Friedlander, 2008].
- Appendix B: *SPGL1 for focal deblending.* This appendix contains a brief discussion of the SPGL1 solver, used to generate the deblending results shown in this thesis.
- Appendix C: *The blending noise operator.* In the last appendix I define the blending noise operator and discuss some of its properties. When applied to unblended data, the blending noise operator produces the blending noise component found in the pseudo-deblended data.

Original contributions found in this work are listed below:

- The formulation of focal deblending (section 3.5).
- The concept of deblending subsets of seismic data using focal transforms and grids tailored to the type of acquisition, for towed streamer acquisitions (section 4.2) and OBN acquisitions (section 4.3).
- The hybrid focal-curvelet deblending strategy (5.3) and associated examples on synthetic data (5.4).
- The demonstrations with field data examples (chapter 6).

Chapters and sections not mentioned in this list are prior work sourced from the cited references, included in this thesis for completeness and as an aid to the reader.

2

Formulation of blending and deblending

2.1 The data cube and notation conventions

For the purposes of this thesis I adopt the convention of representing recorded seismic data as a data cube, i.e. a 3D array of numbers, following closely the conventions introduced in Berkhout [1982]. Let $p_{\text{true}}(\mathbf{r},\, \mathbf{s};\, t)$ be the value of a pressure field, generated by a source at coordinates $\mathbf{s} = (x_s,\, y_s,\, z_s)$ and recorded by a receiver at coordinates $\mathbf{r} = (x_r,\, y_r,\, z_r)$, at time instant t. In a practical acquisition $p_{\text{true}}(\mathbf{r},\, \mathbf{s};\, t)$ is not known everywhere, but only at specific coordinates that are determined by the acquisition geometry. Four useful sets can be defined:

- The set of all unique coordinates where sources were placed, $\mathcal{S} = \{\mathbf{s}_1,\, \mathbf{s}_2,\, \ldots,\, \mathbf{s}_{n_s}\}$, where $n_s = |\mathcal{S}|$ is the number of unique source (or source array) coordinates.

- The set of all unique coordinates where receivers were placed, $\mathcal{R} = \{\mathbf{r}_1,\, \mathbf{r}_2,\, \ldots,\, \mathbf{r}_{n_r}\}$, where $n_r = |\mathcal{R}|$ is the number of unique receiver (or receiver array) coordinates.

- The acquisition set $\mathcal{A}$ of all source-receiver ordered pairs $(\mathbf{r},\, \mathbf{s})$ that were realized in the acquisition.

- The sampling time set $\mathcal{T} = \{0\delta t,\, 1\delta t,\, 2\delta t,\, \ldots,\, (n_t - 1)\delta t\}$, where δt is the time sampling interval and $n_t = |\mathcal{T}|$ is the number of samples recorded at every source-receiver location pair. The assumption is that δt and n_t remain constant for all the acquisition.

More often than not, $\mathcal{A} \neq \mathcal{R} \times \mathcal{S}$, because not all possible pairs of chosen source and receiver coordinates are realized in the actual acquisition.

Using the sets defined above, we can define a new pressure field,

$$p(\mathbf{r},\, \mathbf{s};\, t) = \begin{cases} p_{\text{true}}(\mathbf{r},\, \mathbf{s};\, t), & \text{if } (\mathbf{r},\, \mathbf{s}) \in \mathcal{A} \text{ and } t \in \mathcal{T}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1.1}$$

This effectively means that field values that were not recorded are treated as if they have the value zero. By sampling $p(\mathbf{r},\, \mathbf{s};\, t)$, time slices of the data cube can be constructed. For simplicity I assume that the sources and receivers lie on $x - y$ planes at constant depths z_s and z_r, respectively. Then the time slice at time $t \in \mathcal{T}$ is given by

$$\mathbf{P}(z_r,\, z_s;\, t) = \begin{bmatrix} p(\mathbf{r}_1,\, \mathbf{s}_1;\, t) & p(\mathbf{r}_1,\, \mathbf{s}_2;\, t) & \cdots & p(\mathbf{r}_1,\, \mathbf{s}_{n_s};\, t) \\ p(\mathbf{r}_2,\, \mathbf{s}_1;\, t) & p(\mathbf{r}_2,\, \mathbf{s}_2;\, t) & & \vdots \\ \vdots & & \ddots & \vdots \\ p(\mathbf{r}_{n_r},\, \mathbf{s}_1;\, t) & \cdots & \cdots & p(\mathbf{r}_{n_r},\, \mathbf{s}_{n_s};\, t) \end{bmatrix}, \tag{2.1.2}$$

with

$$\begin{aligned} \mathbf{r}_m &= (x_{r_m},\, y_{r_m},\, z_r), \quad \mathbf{r}_m \in \mathcal{R}, \quad m = 0,\, 1,\, \ldots,\, n_r, \\ \mathbf{s}_n &= (x_{s_n},\, y_{s_n},\, z_s), \quad \mathbf{s}_n \in \mathcal{S}, \quad n = 0,\, 1,\, \ldots,\, n_s. \end{aligned}$$

Note that I construct each time slice by evaluating $p(\mathbf{r},\, \mathbf{s};\, t)$ at every coordinate combination in $\mathcal{R} \times \mathcal{S}$. Because of the way $p(\mathbf{r},\, \mathbf{s};\, t)$ was defined in equation (2.1.1), $\mathbf{P}(z_r,\, z_s;\, t)$ contains zeros for source-receiver combinations that were not realized in the field. $\mathbf{P}(z_r,\, z_s;\, t)$ can accomodate sources and receivers on an $x - y$ plane (3D seismic data), as well as the special case where sources and receivers lie on a line in that plane (2D seismic data). A convention for sorting the samples into $\mathbf{P}(z_r,\, z_s;\, t)$ is to sort first over the y coordinate and then, for each location with the same y coordinate, the samples are sorted over the x coordinate [Kinneging et al., 1989]. A schematic example is shown in figure 2.1.

Many operations on the data cube, such as blending or focusing, will involve convolutions along the time axis. It is computationally advantageous to carry out those operations in the frequency-space (FX), rather than the time-space (TX) domain. In the FX domain the data cube consists of monochromatic frequency slices rather than time slices. A data cube in the TX domain can be easily converted to the FX domain by calculating the discrete Fourier transform (DFT) of each of the traces:

$$\mathbf{P}(z_r,\, z_s;\, \omega) = \sum_{n=0}^{n_t - 1} \mathbf{P}(z_r,\, z_s;\, t) e^{-j\omega n \delta t}, \tag{2.1.3}$$

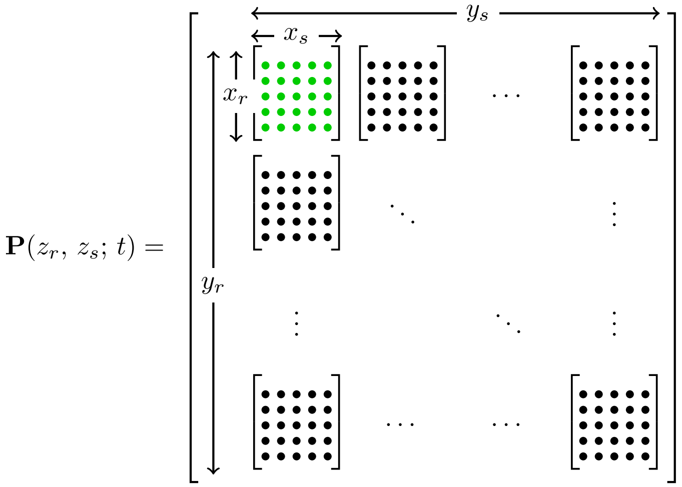


Figure 2.1: *Sorting data samples into a time slice, when the acquisition is done on $x-y$ planes. When the acquisition is on a line, $\mathbf{P}(z_r,\, z_s;\, t)$ consists only of the submatrix $y_r = 0$ m and $y_s = 0$ m (in green), assuming that we have rotated coordinates such that the acquisition line lies on the x-axis.*

with $j = \sqrt{-1}$, $\omega \in \Omega$,

$$\begin{aligned} \Omega &= \{l\delta\omega :\ l_{\min} \le l \le l_{\max}\}, \\ l_{\min} &= -\lfloor (n_t - 1)/2 \rfloor, \\ l_{\max} &= \lfloor n_t/2 \rfloor, \end{aligned} \tag{2.1.4}$$

l integer and $\delta\omega = 2\pi(n_t \delta t)^{-1}$. The DFT is evaluated at a total of n_t discrete frequencies, equal to the number of time slices. The cube can be transformed back to the TX domain via the inverse DFT,

$$\mathbf{P}(z_r,\, z_s;\, t) = \frac{1}{n_t} \sum_{l=l_{\min}}^{l_{\max}} \mathbf{P}(z_r,\, z_s;\, \omega) e^{jl\delta\omega t}. \tag{2.1.5}$$

Often, there is a need to refer to the data cube as a whole, rather than to specific time or frequency slices. To refer to the whole $(n_r \times n_s \times n_t)$ cube, I use the following notation:

- $\hat{\mathbf{P}}(z_r,\, z_s)$ for a data cube that is a concatenation of time slices and
- $\mathbf{P}(z_r,\, z_s)$ for a data cube that is a concatenation of frequency slices.

Several operations can be defined on a data cube. First, I define an additional set of coordinates $\mathcal{U}$, on the plane $z = z_u$, with $n_u = |\mathcal{U}|$, and four auxiliary data cubes:

Data cube	Individual element	Set of receivers	Set of sources	Data cube dimensions
$\mathbf{F}_1(z_r,\, z_s)$	$f_1(\mathbf{r},\, \mathbf{s};\, \omega)$	$\mathcal{R}$	$\mathcal{S}$	$(n_r \times n_s \times n_t)$
$\mathbf{F}_2(z_r,\, z_s)$	$f_2(\mathbf{r},\, \mathbf{s};\, \omega)$	$\mathcal{R}$	$\mathcal{S}$	$(n_r \times n_s \times n_t)$
$\mathbf{F}_3(z_r,\, z_u)$	$f_3(\mathbf{r},\, \mathbf{u};\, \omega)$	$\mathcal{R}$	$\mathcal{U}$	$(n_r \times n_u \times n_t)$
$\mathbf{F}_4(z_u,\, z_s)$	$f_4(\mathbf{u},\, \mathbf{s};\, \omega)$	$\mathcal{U}$	$\mathcal{S}$	$(n_u \times n_s \times n_t)$

The auxiliary data cubes $\mathbf{F}_1$ to $\mathbf{F}_4$ may represent different sets of recorded data or linear operators (e.g. focusing and blending) and focal subdomains, to be introduced later on. For simplicity I am using sets $\mathcal{R}$ and $\mathcal{S}$ to construct the 'source' and 'receiver' dimensions of the cubes in the following examples. Other coordinate sets may play that role too on occasion, depending on what the cube represents, which should be clear from context. An example would be sets of focal gridpoint coordinates acting as the 'source' and 'receiver' dimension in a focal-subdomain data cube.

I list a number of mathematical operations on data cubes, and the condition that must be held for each element $f'(\mathbf{r},\, \mathbf{s};\, \omega)$ of the resulting data cube $\mathbf{F}'(z_r,\, z_s)$, for every $\mathbf{r}$, $\mathbf{s}$, ω.

Conjugation:

$$\mathbf{F}'(z_r,\, z_s) = \mathbf{F}_1(z_r,\, z_s)^* \Rightarrow f'(\mathbf{r},\, \mathbf{s};\, \omega) = f_1(\mathbf{r},\, \mathbf{s};\, \omega)^*.$$

Transposition of frequency slices:

$$\mathbf{F}'(z_s,\, z_r) = \mathbf{F}_1(z_r,\, z_s)^{\mathrm{T}} \Rightarrow f'(\mathbf{s},\, \mathbf{r};\, \omega) = f_1(\mathbf{r},\, \mathbf{s};\, \omega).$$

Conjugate (Hermitian) transposition of frequency slices:

$$\mathbf{F}'(z_s,\, z_r) = \mathbf{F}_1(z_r,\, z_s)^{\mathrm{H}} \Rightarrow f'(\mathbf{s},\, \mathbf{r};\, \omega) = f_1(\mathbf{r},\, \mathbf{s};\, \omega)^*.$$

Addition:

$$\begin{gathered}\mathbf{F}'(z_r,\, z_s) = \mathbf{F}_1(z_r,\, z_s) + \mathbf{F}_2(z_r,\, z_s) \Rightarrow \\ f'(\mathbf{r},\, \mathbf{s};\, \omega) = f_1(\mathbf{r},\, \mathbf{s};\, \omega) + f_2(\mathbf{r},\, \mathbf{s};\, \omega).\end{gathered}$$

Multiplication with a scalar β:

$$\begin{gathered}\mathbf{F}'(z_r,\, z_s) = \beta\mathbf{F}_1(z_r,\, z_s) \Rightarrow \\ f'(\mathbf{r},\, \mathbf{s};\, \omega) = \beta f_1(\mathbf{r},\, \mathbf{s};\, \omega).\end{gathered}$$

Slice-by-slice multiplication:

$$\mathbf{F}'(z_r,\, z_s) = \mathbf{F}_3(z_r,\, z_u)\mathbf{F}_4(z_u,\, z_s) \Rightarrow$$
$$f'(\mathbf{r},\, \mathbf{s};\, \omega) = \sum_{\mathbf{u}} f_3(\mathbf{r},\, \mathbf{u};\, \omega) f_4(\mathbf{u},\, \mathbf{s};\, \omega).$$

I also define two norms that will be frequently used in the following chapters, namely the Frobenius norm,

$$||\mathbf{F}(z_r,\, z_s)||_{\mathrm{F}} = \sqrt{\sum_{\mathbf{r}} \sum_{\mathbf{s}} \sum_{\omega} |f(\mathbf{r},\, \mathbf{s};\, \omega)|^2},$$

and the sum norm,

$$||\mathbf{F}(z_r,\, z_s)||_{\mathrm{S}} = \sum_{\mathbf{r}} \sum_{\mathbf{s}} \sum_{\omega} |f(\mathbf{r},\, \mathbf{s};\, \omega)|\,.$$

The Frobenius and sum norms corresponds to the ℓ_2- and ℓ_1-norm of the vectorized data cube, respectively. The operations and norms listed above are defined in an analogous way for data cubes in the TX domain.

$\mathbf{P}(z_r,\, z_s;\, t)$ is built with a certain structure, because of the way its elements are organized. As can be seen in equation (2.1.2), each column of a slice is associated with a unique source coordinate; each row with a unique receiver coordinate and each element with a pair of unique source-receiver coordinates. Below I define a few data cube subsets that I use often in the discussions in this thesis.

- The *trace*: a trace is formed by extracting the element pertaining to the source-receiver pair $(\mathbf{r}_m,\, \mathbf{s}_n)$ from all time slices, and then concatenating the individual elements into a vector (figure 2.2a).

- The *common shot gather*: a common shot gather is formed by extracting the column pertaining to a source at $\mathbf{s}_n$ from all time slices, and then concatenating the individual columns into a matrix (figure 2.2b).

- The *common receiver gather*: a common receiver gather is formed by extracting the row pertaining to a receiver at $\mathbf{r}_m$ from all time slices, and then concatenating the individual rows into a matrix (figure 2.2c). For towed streamer acquisition a complication arises due to the fact that the receivers are not stationary and their locations are not repeatable exactly. This can lead to common receiver gathers with very few traces. A partial remedy is to treat receivers within a small radius as having the same location, at the cost of introducing traveltime errors.

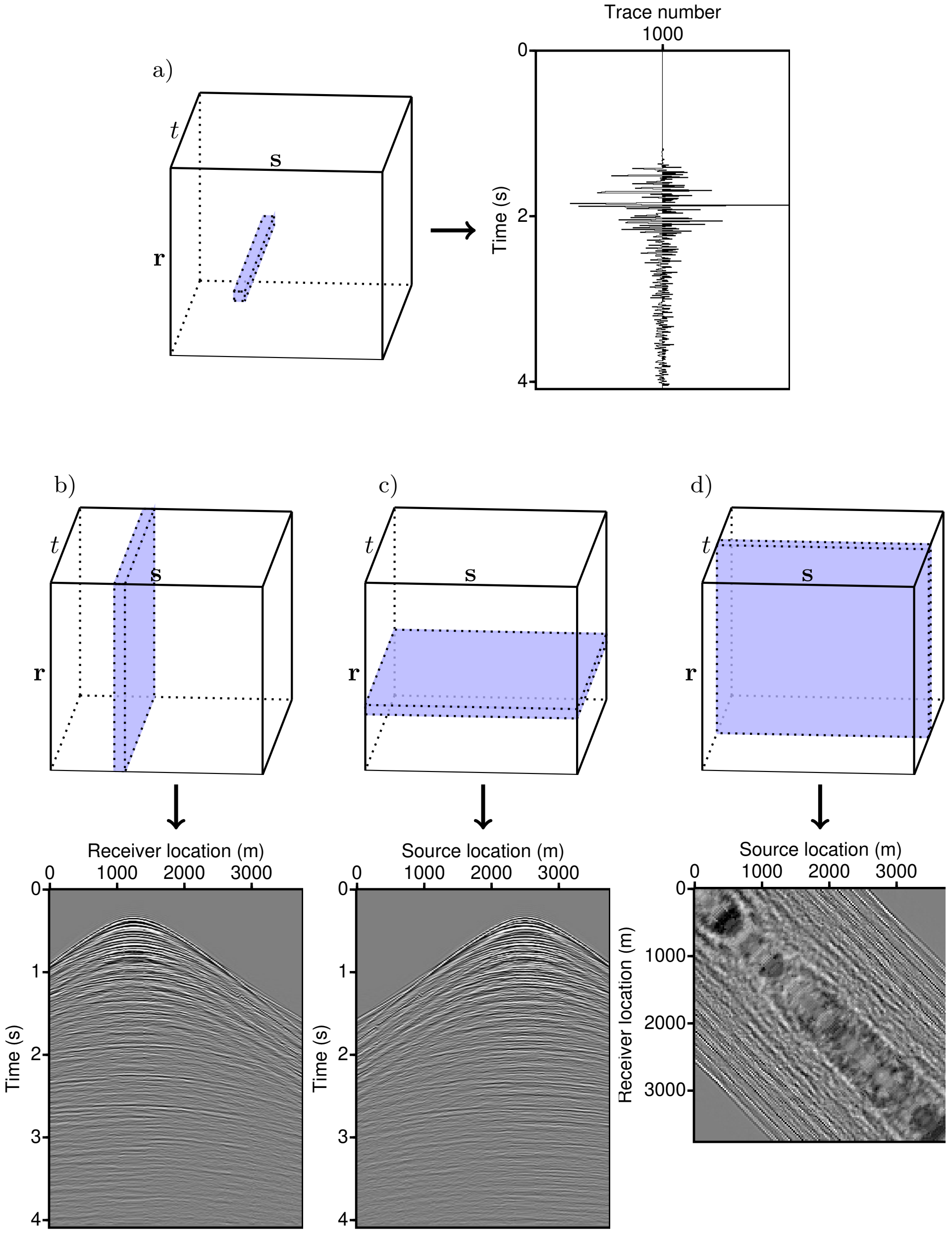


Figure 2.2: *Examples of data data cube subsets; a) a seismic trace, b) a common shot gather, c) a common receiver gather, d) a time slice.*

2.2 Blending

As described in chapter 1, we refer to the situation where multiple sources fire within the same recording window as blending. Mathematically, blending can be seen as a two-step process. The first step is to encode each shot gather with the code associated with the corresponding source. The second step is to create *blended* shot gathers, by stacking (summing) encoded common shot gathers. Some parameters of the code may not be predetermined, e.g. when using a random time delay code, the actual time delays may be known only after the acquisition has taken place. What is important is that, after acquisition, all parameters of the blending code should be known as accurately as possible. This is especially true for inversion-based methods, such as the one discussed in this thesis.

The code may assume many different forms, but due to acquisition limitations, not all of them are physically realizable for field acquisition. Here I restrict the discussion to codes that are (circularly)[1] convolutional in time, i.e. an encoded trace is created by (circularly) convolving an unencoded trace with a coding signal. Most practical blending codes belong to this category.

The convolution theorem for the DFT, then, dictates that encoding can be carried out in the FX domain by elementwise multiplying each trace with the transformed coding signal. Let $\gamma(\mathbf{s},\, b;\, \omega)$ be a frequency sample of the code used to encode a shot at $\mathbf{s}$ when that shot is used for generating the blended shot gather b. Then, the blended trace sample at angular frequency ω recorded by a receiver at $\mathbf{r}$ is given by

$$p_{\mathrm{bl}}(\mathbf{r},\, b;\, \omega) = \sum_{\mathbf{s}} p(\mathbf{r},\, \mathbf{s};\, \omega)\gamma(\mathbf{s},\, b;\, \omega). \tag{2.2.6}$$

The sum runs over all shot locations, but we can easily exclude shots from blended gathers by setting $\gamma(\mathbf{s},\, b;\, \omega) = 0$ for the corresponding combinations of $\mathbf{s}$ and b. Also note that $p_{\mathrm{bl}}(\mathbf{r},\, b;\, \omega)$ is not associated with any single shot coordinate anymore; to highlight this I replace the shot coordinate with an integer $b \in \mathcal{B}$, with $\mathcal{B} = \{1,\, 2,\, \ldots,\, n_{\mathrm{bl}}\}$, that simply indexes the blended shot gather. The number of recorded blended shot gathers is given by n_{bl}. Note that for $p_{\mathrm{bl}}(\mathbf{r},\, b;\, t)$ to be purely real for every $t \in \mathcal{T}$,

$$\gamma(\mathbf{s},\, b;\, \omega) = \gamma(\mathbf{s},\, b;\, -\omega)^*, \quad \omega \in \Omega, \tag{2.2.7}$$

when n_t is odd. When n_t is even and the Nyquist frequency is present,

$$\begin{aligned} \gamma(\mathbf{s},\, b;\, \omega) &= \gamma(\mathbf{s},\, b;\, -\omega)^* \quad && \omega \in \Omega\backslash\{\lfloor n_t/2 \rfloor \delta\omega\}, \\ \gamma(\mathbf{s},\, b;\, \omega) &= \gamma(\mathbf{s},\, b;\, \omega)^* \quad && \omega = \lfloor n_t/2 \rfloor \delta\omega, \end{aligned} \tag{2.2.8}$$

---

[1] Circular convolution can be converted to linear convolution by zero-padding traces, circularly convolving them, and truncating the result.

or, in other words, the Nyquist component must not have an imaginary part, if present. When conditions (2.2.7) or (2.2.8) are met, $\gamma(\mathbf{s},\, b;\, \omega)$ satisfies the conjugate symmetry of the DFT.

A number of familiar codes can be implemented by defining $\gamma(\mathbf{s},\, b;\, \omega)$ in different ways:

- *Shot repetition codes*: these are codes where a shot is repeated n_{rep} times, $n_{\text{rep}} > 1$. Then,

  $$\gamma(\mathbf{s},\, b;\, \omega) = \sum_{i=1}^{n_{\text{rep}}} \gamma_i(\mathbf{s},\, b;\, \omega),$$

  where each repeated shot i is assigned code $\gamma_i(\mathbf{s},\, b;\, \omega)$. Usually, $\gamma_i(\mathbf{s},\, b;\, \omega)$ takes the form of a delay code (discussed below), where the delays are optimized for code orthogonality.

- *Amplitude-phase codes*: these codes are parameterized by the amplitude of the source, $a(\mathbf{s},\, b)$, and a phase function, $\phi(\mathbf{s},\, b;\, \omega)$, such that

  $$\gamma(\mathbf{s},\, b;\, \omega) = a(\mathbf{s},\, b)e^{j\phi(\mathbf{s},\, b;\, \omega)}.$$

  All the codes listed below are special cases of the amplitude-phase code.

- *Frequency sweep codes*: a specialization of amplitude-phase codes where the instantaneous frequency changes according to a pattern. An example of this is the tapered linear frequency sweep code with

  $$a(\mathbf{s},\, b) = \begin{cases} \frac{\alpha_0(\mathbf{s},b)}{2\delta t}\sqrt{\frac{2\pi T_{\text{sw}}(\mathbf{s},b)}{\omega_2(\mathbf{s},b)-\omega_1(\mathbf{s},b)}}, & \text{if the shot } \mathbf{s} \text{ is present in blended gather } b, \\ 0, & \text{otherwise,} \end{cases}$$

  $$\phi(\mathbf{s},\, b;\, \omega) = -\frac{T_{\text{sw}}(\mathbf{s},b)}{2}\frac{\omega^2-\omega_1^2(\mathbf{s},b)}{\omega_2(\mathbf{s},b)-\omega_1(\mathbf{s},b)}.$$

  $T_{\text{sw}}(\mathbf{s},b)$ is the sweep length, $\omega_1(\mathbf{s},b)$ is the starting frequency, $\omega_2(\mathbf{s},b)$ is the ending frequency and $\alpha_0(\mathbf{s},b)$ the source amplitude [Meunier, 2011].

- *Time delay codes*: a specialization of amplitude-phase codes, with

  $$a(\mathbf{s},\, b) = \begin{cases} 1, & \text{if the shot } \mathbf{s} \text{ is present in blended gather } b, \\ 0, & \text{otherwise,} \end{cases}$$

  $$\phi(\mathbf{s},\, b;\, \omega) = -\omega\tau_d(\mathbf{s},\, b), \tag{2.2.9}$$

  $\tau_d(\mathbf{s},\, b)$ being a time delay. This code shifts traces in time because phase is proportional to frequency.

- *Random time delay codes*: time delay codes where $\tau_d(\mathbf{s},\, b)$ is random.

- *Apparition codes*: time delay codes where $\tau_d(\mathbf{s},\, b)$ follows a pattern that alternates between fixed delay values. The periodic pattern modulates the blending noise such that part of it can be perfectly removed [Robertsson et al., 2016a]. Apparition codes are also possible to implement using amplitude modulation.

This thesis will focus mostly on random time delay codes, with the additional constraint that each shot should participate in only one blended shot gather; therefore

$$\sum_b a(\mathbf{s},\, b) = 1, \quad \forall \mathbf{s} \in \mathcal{S}, \tag{2.2.10}$$

keeping in mind that $a(\mathbf{s},\, b) \in \{0,\, 1\}$ for purely delay-based codes. This extra constraint in code structure will be there to make blending codes more realistic, as it is rather unlikely that the same common shot gather will be acquired more than once so that it can contribute to more than one blended gathers. This is especially true for marine acquisition, where the sources are in constant motion.

Blending, as defined in equation (2.2.6) is a linear operation that can be implemented as multiplication of frequency slices of the unblended data with a blending operator $\mathbf{\Gamma}(z_s)$ [Berkhout, 2008], that has the same form as a data cube. Using the notation for slice-by-slice multiplication,

$$\mathbf{P}_{\mathrm{bl}}(z_r,\, z_s) = \mathbf{P}(z_r,\, z_s)\mathbf{\Gamma}(z_s), \tag{2.2.11}$$

with slices

$$\mathbf{P}_{\mathrm{bl}}(z_r,\, z_s;\, \omega) = \begin{bmatrix} p_{\mathrm{bl}}(\mathbf{r}_1,\, 1;\, \omega) & p_{\mathrm{bl}}(\mathbf{r}_1,\, 2;\, \omega) & \cdots & p_{\mathrm{bl}}(\mathbf{r}_1,\, n_{\mathrm{bl}};\, \omega) \\ p_{\mathrm{bl}}(\mathbf{r}_2,\, 1;\, \omega) & p_{\mathrm{bl}}(\mathbf{r}_2,\, 2;\, \omega) & & \vdots \\ \vdots & & \ddots & \vdots \\ p_{\mathrm{bl}}(\mathbf{r}_{n_r},\, 1;\, \omega) & \cdots & \cdots & p_{\mathrm{bl}}(\mathbf{r}_{n_r},\, n_{\mathrm{bl}};\, \omega) \end{bmatrix},$$

and

$$\mathbf{\Gamma}(z_s;\, \omega) = \begin{bmatrix} \gamma(\mathbf{s}_1,\, 1;\, \omega) & \gamma(\mathbf{s}_1,\, 2;\, \omega) & \cdots & \gamma(\mathbf{s}_1,\, n_{\mathrm{bl}};\, \omega) \\ \gamma(\mathbf{s}_2,\, 1;\, \omega) & \gamma(\mathbf{s}_2,\, 2;\, \omega) & & \vdots \\ \vdots & & \ddots & \vdots \\ \gamma(\mathbf{s}_{n_s},\, 1;\, \omega) & \cdots & \cdots & \gamma(\mathbf{s}_{n_s},\, n_{\mathrm{bl}};\, \omega) \end{bmatrix}, \tag{2.2.12}$$

Any solution of the deblending problem must satisfy equation (2.2.11). The blending operation for monochromatic frequency slices is depicted in figure 2.3, for the case where each blended record is formed by summing two unblended shot records.

For a blended dataset, we can make the following observations regarding the coherency of seismic events when random time delay codes are used:

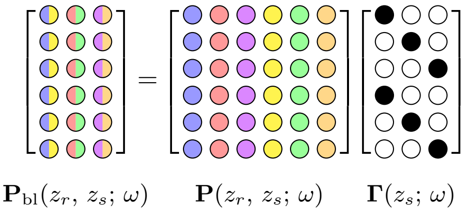


Figure 2.3: *Schematic representation of the blending operation for a single monochromatic frequency slice. Six common shot gathers are blended to produce three blended shot gathers.*

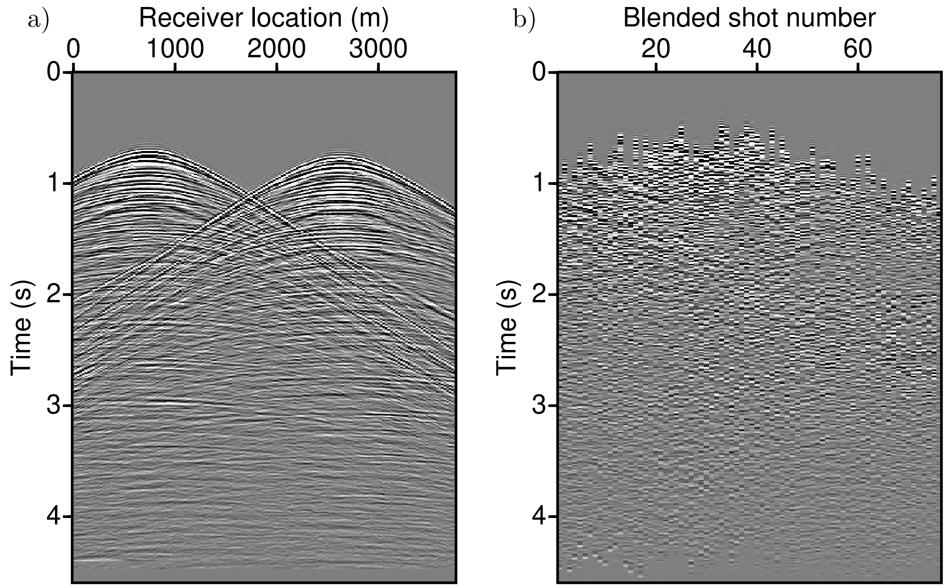


Figure 2.4: *Two gathers extracted from a blended dataset. a) A blended shot gather, b) a blended common receiver gather. A time delay code was used for blending.*

- In blended shot gathers (figure 2.4a), all events are coherent. This happens because all traces of each contributing unblended gather have been shifted in time by the same amount.

- In blended common receiver gathers (figure 2.4b), there is no event coherency. Every blended trace is the sum of traces related to different sources, which will be assigned different time shifts, in general.

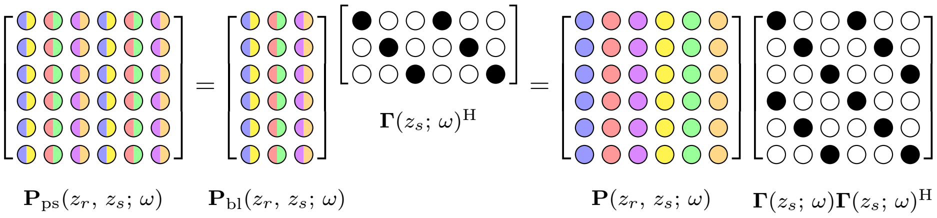


Figure 2.5: *Schematic representation of the pseudo-deblending operation for a single monochromatic frequency slice. Three blended shot gathers expand into six pseudo-deblended common shot gathers.*

2.3 Pseudo-deblending

Pseudo-deblending is the adjoint of the blending operation. Mathematically, it is also implemented as a matrix-matrix multiplication of frequency slices of the blended data and the Hermitian of the blending operator,

$$\begin{aligned}\mathbf{P}_{\mathrm{ps}}(z_r,\,z_s) &= \mathbf{P}_{\mathrm{bl}}(z_r,\,z_s)\mathbf{\Gamma}(z_s)^{\mathrm{H}}\\ &= \mathbf{P}(z_r,\,z_s)\mathbf{\Gamma}(z_s)\mathbf{\Gamma}(z_s)^{\mathrm{H}}.\end{aligned} \tag{2.3.13}$$

Pseudo-deblending can be also viewed as a two-step process. The first step is to generate as many copies of a blended shot gather as the number of sources that are blended in it (figure 2.5). This step expands each $(n_r \times n_{\mathrm{bl}})$ frequency slice into a $(n_r \times n_s)$ pseudo-deblended frequency slice, that has the same dimensions as unblended data. The second step is decoding, in which every common shot gather is multiplied with the complex conjugate of the coding signal associated with the source.

Several useful remarks can be made about the behavior of the chosen code, by examining the relationship of a trace of unblended data, with the corresponding pseudo-deblended trace. From equation (2.3.13) we get

$$p_{\mathrm{ps}}(\mathbf{r},\,\mathbf{s};\,\omega) = \sum_b p_{\mathrm{bl}}(\mathbf{r},\,b;\,\omega)\gamma(\mathbf{s},\,b;\,\omega)^*. \tag{2.3.14}$$

for an individual frequency sample of a trace. Plugging equation (2.2.6) into (2.3.14) we get

$$\begin{aligned}p_{\mathrm{ps}}(\mathbf{r},\,\mathbf{s};\,\omega) &= \sum_b \left(\sum_{\mathbf{s}'} p(\mathbf{r},\,\mathbf{s}';\,\omega)\gamma(\mathbf{s}',\,b;\,\omega)\right)\gamma(\mathbf{s},\,b;\,\omega)^* &\quad (2.3.15)\\ &= \sum_{\mathbf{s}'} p(\mathbf{r},\,\mathbf{s}';\,\omega)\sum_b \gamma(\mathbf{s}',\,b;\,\omega)\gamma(\mathbf{s},\,b;\,\omega)^*, &\quad (2.3.16)\end{aligned}$$

with $\mathbf{s}' \in \mathcal{S}$. Separating the shot at $\mathbf{s}$ from the sum yields

$$\begin{aligned} p_{\mathrm{ps}}(\mathbf{r},\, \mathbf{s}; \omega) &= p(\mathbf{r},\, \mathbf{s};\, \omega) \sum_b |\gamma(\mathbf{s},\, b;\, \omega)|^2 + \\ &+ \sum_{\mathbf{s}' \neq \mathbf{s}} p(\mathbf{r},\, \mathbf{s}';\, \omega) \sum_b \gamma(\mathbf{s}',\, b;\, \omega)\gamma(\mathbf{s},\, b;\, \omega)^* \\ &= \underbrace{p(\mathbf{r},\, \mathbf{s};\, \omega)c(\mathbf{s},\, \mathbf{s};\, \omega)}_{\text{Signal (scaled)}} + \underbrace{\sum_{\mathbf{s}' \neq \mathbf{s}} p(\mathbf{r},\, \mathbf{s}';\, \omega)c(\mathbf{s}',\, \mathbf{s};\, \omega)}_{\text{Blending noise (scaled)}}, \end{aligned} \tag{2.3.17}$$

where

- $c(\mathbf{s},\, \mathbf{s};\, \omega) = \sum_b |\gamma(\mathbf{s},\, b;\, \omega)|^2$ is a frequency sample from the sum of the autocorrelations of the coding signals associated with shot location $\mathbf{s}$ and
- $c(\mathbf{s}',\, \mathbf{s};\, \omega) = \sum_b \gamma(\mathbf{s}',\, b;\, \omega)\gamma(\mathbf{s},\, b;\, \omega)^*$ is a frequency sample from the sum of the crosscorrelations of the coding signals associated with shot locations $\mathbf{s}$ and $\mathbf{s}'$.

Assuming $|c(\mathbf{s},\, \mathbf{s};\, \omega)| \neq 0$, we can normalize $p_{\mathrm{ps}}(\mathbf{r},\, \mathbf{s}; \omega)$ such that the signal part is at the same scale as the unblended data,

$$\frac{1}{c(\mathbf{s},\, \mathbf{s};\, \omega)} p_{\mathrm{ps}}(\mathbf{r},\, \mathbf{s}; \omega) = \underbrace{p(\mathbf{r},\, \mathbf{s};\, \omega)}_{\text{Signal}} + \underbrace{\sum_{\mathbf{s}' \neq \mathbf{s}} p(\mathbf{r},\, \mathbf{s}';\, \omega) \frac{c(\mathbf{s}',\, \mathbf{s};\, \omega)}{c(\mathbf{s},\, \mathbf{s};\, \omega)}}_{\text{Blending noise (scaled)}}. \tag{2.3.18}$$

The ratio $c(\mathbf{s}',\, \mathbf{s};\, \omega)/c(\mathbf{s},\, \mathbf{s};\, \omega)$, which depends solely on the blending code, controls the amount of contamination present in the pseudo-deblended sample $p_{\mathrm{ps}}(\mathbf{r},\, \mathbf{s}; \omega)$. A desirable property of a blending code is to keep this ratio as close to zero as possible. When the ratio is low for all samples, most of the blending noise vanishes just by applying the pseudo-deblending operation. When designing the code, the case $|c(\mathbf{s},\, \mathbf{s};\, \omega)| = 0$ should be avoided, as it has the effect of removing frequency components from the spectrum of the corresponding trace.

Since a random time delay code will be used for most of the examples in this thesis, it is interesting to see how it fares with respect to blending noise suppression. Using $\gamma(\mathbf{s},\, b;\, \omega) = a(\mathbf{s},\, b)e^{-j\omega\tau_d(\mathbf{s},\, b)}$ we get

$$\frac{c(\mathbf{s}',\, \mathbf{s};\, \omega)}{c(\mathbf{s},\, \mathbf{s};\, \omega)} = \frac{\sum_b a(\mathbf{s}',\, b)a(\mathbf{s},\, b)e^{-j\omega(\tau_d(\mathbf{s}',\, b) - \tau_d(\mathbf{s},\, b))}}{\sum_b a(\mathbf{s},\, b)^2}. \tag{2.3.19}$$

Using the assumption that each shot gather contributes to only one blended shot gather, equation (2.2.10) holds for both $\mathbf{s}$ and $\mathbf{s}'$. This means that there can be at most one element $b' \in \mathcal{B}$ for which $a(\mathbf{s}', b') = a(\mathbf{s}, b') = 1$. Also, because both $a(\mathbf{s}, b) \in \{0, 1\}$ and equation (2.2.10) hold, $\sum_b a(\mathbf{s}, b)^2 = \sum_b a(\mathbf{s}, b) = 1$. Given these,

$$\frac{c(\mathbf{s}', \mathbf{s}; \omega)}{c(\mathbf{s}, \mathbf{s}; \omega)} = \begin{cases} e^{-j\omega(\tau_d(\mathbf{s}', b) - \tau_d(\mathbf{s}, b))}, & \text{when } b = b' \\ 0, & \text{otherwise.} \end{cases} \tag{2.3.20}$$

The magnitude of the ratio is then

$$\frac{|c(\mathbf{s}', \mathbf{s}; \omega)|}{|c(\mathbf{s}, \mathbf{s}; \omega)|} = \begin{cases} 1, & \text{when } b = b' \\ 0, & \text{otherwise.} \end{cases} \tag{2.3.21}$$

Two important conclusions about delay codes can be drawn from this analysis.

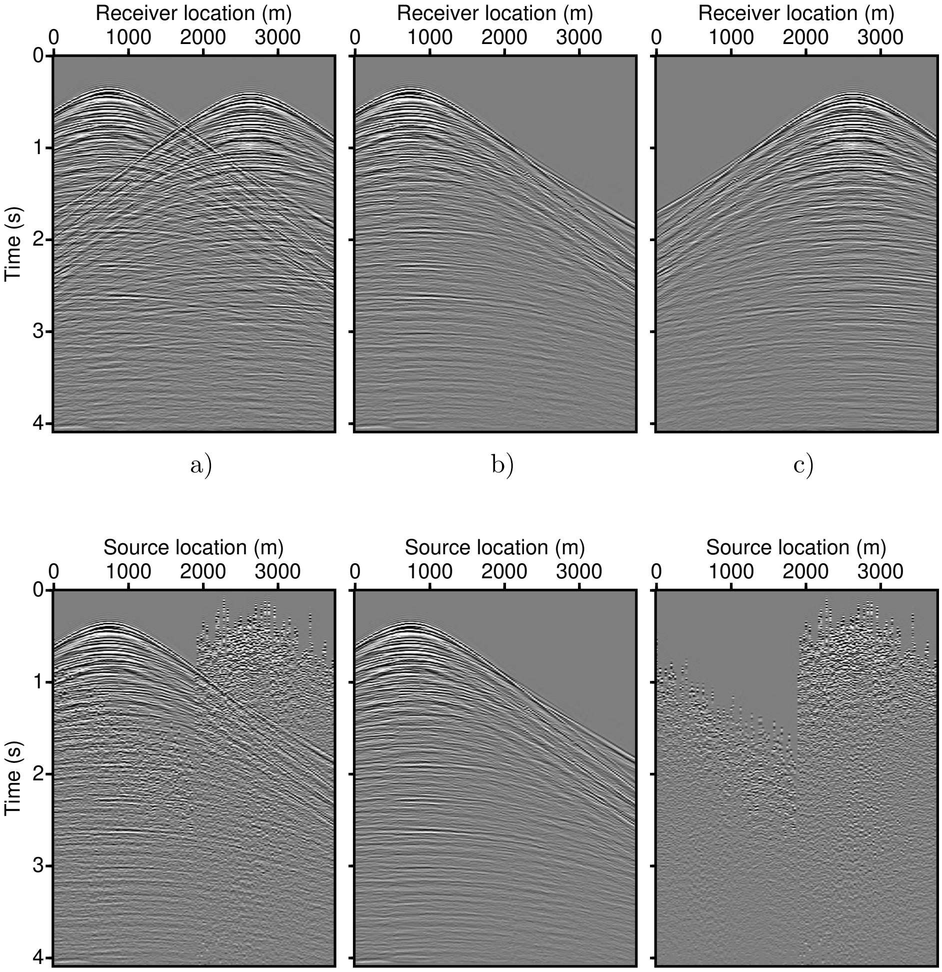
- There will be blending noise present in the pseudo-deblended data cube. The noise traces will be time-shifted by an amount dictated by the residual delays $\tau_d(\mathbf{s}', b) - \tau_d(\mathbf{s}, b)$. This is a consequence of equations (2.3.18) and (2.3.20).
- Pseudo-deblending will *not* attenuate blending noise, for simple delay codes. This, in turn, means that the only help a deblending algorithm gets from this type of code are the residual delays. This is a consequence of equation (2.3.21).

Signal and noise in pseudo-deblended data have different properties in different seismic gathers, when a random time delay code is used.

- In common shot gathers (figures 2.6a-c), both signal and blending noise are coherent.The signal part will exhibit zero time delay and the noise part a time delay that is constant for all traces.
- In common receiver gathers (figures 2.6d-f), the signal is coherent, but blending noise is not.

In general, the signal will be coherent in all types of gathers, because all time delay has been removed by the pseudo-deblending operation. Noise, on the other hand, will have a different residual time delay for each source, therefore, it will be incoherent for all types of gathers, except the common shot gather. This incoherency introduced by the code will be exploited later in the deblending process.

The code can have a very big influence in the amount of effort that needs to be spent on deblending. This will be shown using a blending code that, while entirely unrealistic for seismic data acquisition purposes, is illustrative of what is possible, at least mathematically. For this blending scheme, pairs of shot gathers will be blended together. The two shot gathers in the pair will be encoded using Gold codes $\mathbf{g}_1$, $\mathbf{g}_2$, respectively, with



a) b) c)

d) e) f)

Figure 2.6: *Comparison of unblended and pseudo-deblended data. Upper row: common shot gather of a) pseudo-deblended data, b) unblended data, c) blending noise. Lower row: common receiver gather of d) pseudo-deblended data, e) unblended data, f) blending noise.*

$$\begin{aligned}\mathbf{g}_1 = [&-1,-1,-1,-1,-1,-1,-1,+1,-1,-1,-1,+1,+1,-1,+1,+1,\\&-1,-1,-1,-1,+1,+1,-1,-1,+1,+1,+1,-1,-1,+1,+1]^{\mathrm{T}},\\\mathbf{g}_2 = [&+1,+1,-1,-1,-1,+1,+1,+1,+1,+1,+1,+1,-1,-1,-1,+1,\\&-1,-1,-1,+1,+1,+1,+1,-1,-1,-1,+1,-1,+1,-1,-1]^{\mathrm{T}}.\end{aligned}$$

Such codes are commonly used in telecommunications to distinguish the transmissions of individual users broadcasting on the same channel [Rao and Dianat, 2005]. Here blending will be done in a different manner than what has been described so far in this chapter. In this case it is easier to describe the blending process in terms of each individual sample, rather than for the data cube as a whole. Blending two samples $p(\mathbf{r},\, \mathbf{s}_1;\, t)$ and $p(\mathbf{r},\, \mathbf{s}_2;\, t)$ yields a vector of samples

$$\mathbf{p}_{\mathrm{bl}}(\mathbf{r},\, b;\, t) = p(\mathbf{r},\, \mathbf{s}_1;\, t)\mathbf{g}_1 + p(\mathbf{r},\, \mathbf{s}_2;\, t)\mathbf{g}_2.$$

Pseudo-deblending a sample can be done by calculating an inner product of the corresponding Gold code and the blended vector of samples

$$p_{\mathrm{ps}}(\mathbf{r},\, \mathbf{s}_1;\, t) = \mathbf{g}_1^{\mathrm{T}}\mathbf{p}_{\mathrm{bl}}(\mathbf{r},\, b;\, t) = p(\mathbf{r},\, \mathbf{s}_1;\, t)\mathbf{g}_1^{\mathrm{T}}\mathbf{g}_1 + p(\mathbf{r},\, \mathbf{s}_2;\, t)\mathbf{g}_1^{\mathrm{T}}\mathbf{g}_2.$$

Scaling the pseudo-deblended sample by $1/\mathbf{g}_1^{\mathrm{T}}\mathbf{g}_1$ yields

$$\frac{1}{\mathbf{g}_1^{\mathrm{T}}\mathbf{g}_1}p_{\mathrm{ps}}(\mathbf{r},\, \mathbf{s}_1;\, t) = p(\mathbf{r},\, \mathbf{s}_1;\, t) + p(\mathbf{r},\, \mathbf{s}_2;\, t)\frac{\mathbf{g}_1^{\mathrm{T}}\mathbf{g}_2}{\mathbf{g}_1^{\mathrm{T}}\mathbf{g}_1}.$$

It is easy to verify that $\mathbf{g}_1^{\mathrm{T}}\mathbf{g}_2/\mathbf{g}_1^{\mathrm{T}}\mathbf{g}_1 = 1/31$, therefore the blending noise will be attenuated by a factor of 31, simply by applying the pseudo-deblending operation (figures 2.7a-c). Depending on the amount of blending noise that can be tolerated for the next stages of processing, the pseudo-deblended data may not need any further deblending. This was a toy example, but it demonstrates the important effect that the blending code can have on the deblending effort.

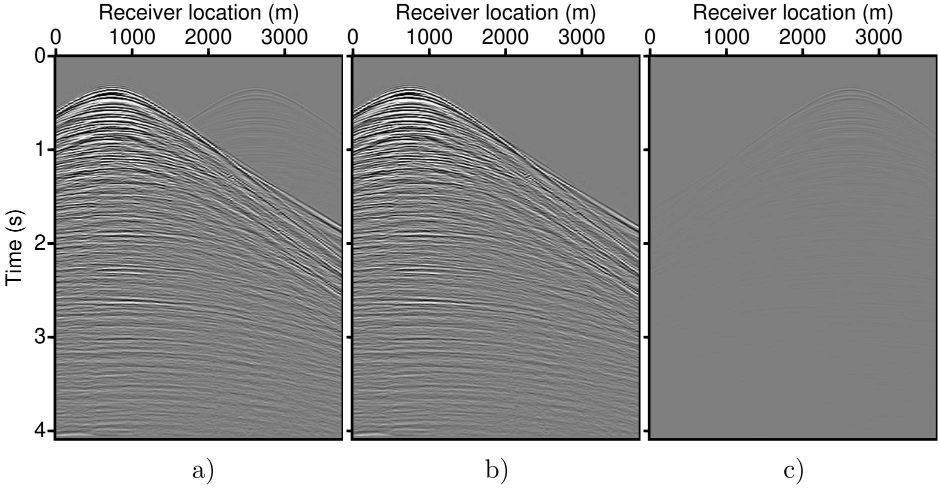
2.4 Deblending as an inverse problem

In the ideal situation, a deblended data cube $\mathbf{P}_{\mathrm{debl}}(z_r,\, z_s)$ should satisfy the blending equation (2.2.11), therefore

$$\mathbf{P}_{\mathrm{bl}}(z_r,\, z_s) = \mathbf{P}_{\mathrm{debl}}(z_r,\, z_s)\mathbf{\Gamma}(z_s) \tag{2.4.22}$$

should hold. Solving for the deblended data cube is usually casted as an optimization problem of the general form

$$\mathbf{P}_{\mathrm{debl}}(z_r,\, z_s) = \underset{\mathbf{P}(z_r,\, z_s)}{\arg\min}\left\{\mathrm{dist}\left(\mathbf{P}_{\mathrm{bl}}(z_r,\, z_s),\, \mathbf{P}(z_r,\, z_s)\mathbf{\Gamma}(z_s)\right)\right\}, \tag{2.4.23}$$



a) b) c)

Figure 2.7: *Comparison of unblended and pseudo-deblended data (using Gold codes for blending). Common shot gather of a) the (scaled) pseudo-deblended data, b) the unblended data, c) the blending noise.*

where dist $(\cdot, \cdot)$ is a chosen distance function. The main difficulty of solving (2.4.23) is that the solution is non-unique when $n_{\text{bl}} < n_s$. We can associate a null space with the blending operation, i.e. the set

$$\mathcal{N}(\mathbf{\Gamma}(z_s)) = \{\mathbf{P}_{\text{null}}(z_r, z_s) \ : \ \mathbf{P}_{\text{null}}(z_r, z_s)\mathbf{\Gamma}(z_s) = \mathbf{0}_{n_r \times n_{\text{bl}} \times n_t}\}, \tag{2.4.24}$$

where $\mathbf{0}_{n_r \times n_{\text{bl}} \times n_t}$ is a $(n_r \times n_{\text{bl}} \times n_t)$ data cube of zeros. If $n_{\text{bl}} = n_s$ and each frequency slice of $\mathbf{\Gamma}(z_s)$ is full rank, $\mathbf{P}_{\text{null}}(z_r, z_s) = \mathbf{0}_{n_r \times n_{\text{bl}} \times n_t}$ and there is a unique solution. In this hypothetical scenario each frequency slice of $\mathbf{\Gamma}(z_s)$ is invertible and deblending amounts to applying the inverse to the corresponding frequency slice of the blended data. For a realistic blended acquisition, however, what would have been two or more independent shot gathers in the corresponding unblended acquisition, contract into one blended shot gather. Therefore, for field blending, $n_{\text{bl}} < n_s$. This implies that there exist infinite nonzero data cubes $\mathbf{P}_{\text{null}}(z_r, z_s) \in \mathcal{N}(\mathbf{\Gamma}(z_s))$ for which $\mathbf{P}_{\text{debl}}(z_r, z_s) + \mathbf{P}_{\text{null}}(z_r, z_s)$ is a valid solution to (2.4.23), if $\mathbf{P}_{\text{debl}}(z_r, z_s)$ is also a valid solution. It is worth noting that, for phase-only codes where each source participates in only one blended gather, a scaled version of $\mathbf{P}_{\text{ps}}(z_r, z_s)$, is part of the solution set as well [Mahdad et al., 2011].

The purpose of deblending is to recover $\mathbf{P}_{\text{debl}}(z_r, z_s) = \mathbf{P}_{\text{ideal}}(z_r, z_s)$. $\mathbf{P}_{\text{ideal}}(z_r, z_s)$ is the 'ground truth', or, in other words, the data that would have been recorded in the ideal case of no blending. Clearly, when $n_{\text{bl}} < n_s$, knowledge of the blended data and the blending code alone does not suffice to guarantee recovery of $\mathbf{P}_{\text{ideal}}(z_r, z_s)$.

Restricting the set of solutions can be done by adding constraints to (2.4.23), and/or by using some form of regularization. This is a way to bring additional information into the problem, that hopefully steers the optimization to a solution that contains minimal blending noise leakage. What separates the different inversion-based methods is the choice of the distance function and the extra constraints or regularization terms they introduce into the optimization problem.

For numerical blending experiments, $\hat{\mathbf{P}}_{\text{ideal}}(z_r,\, z_s)$ is known beforehand, which makes it possible to quantify the quality of the deblended result. For this purpose, I adopt the metric

$$Q_{\text{dB}} = 10 \log_{10} \frac{\left|\left|\hat{\mathbf{P}}_{\text{ideal}}(z_r,\, z_s)\right|\right|_{\text{F}}^2}{\left|\left|\hat{\mathbf{P}}_{\text{debl}}(z_r,\, z_s) - \hat{\mathbf{P}}_{\text{ideal}}(z_r,\, z_s)\right|\right|_{\text{F}}^2}. \tag{2.4.25}$$

3

Focal deblending

An integral part of focal deblending is the focal transform. The focal transform uses an approximate velocity model of the subsurface and wavefield redatuming to transform a dataset recorded at the surface into a collection of datasets approximately focused at chosen depth levels. The focusing operation allows us to implicitly take advantage of the coherency of the signal to be extracted and, with the help of a sparsity-promoting objective function, discriminate against incoherent blending noise. The focusing process is based on an NMO velocity model.

In this chapter two variants of the transform are examined. The first is the double focal transform, in which scattering and propagation are separated. Berkhout and Verschuur [2010] used this version of the transform for data denoising. Kutscha and Verschuur [2012] later paired it with parsimonious optimization for seismic data interpolation. The single-sided focal transform is the second variant. It uses two-way operators that combine both scattering and propagation. It is based on an earlier definition of the focal transform by Berkhout and Verschuur [2006] used for denoising prestack data.

Both variants of the focal transform are conceptually linked to a simplified version of WRW[1] modeling, commonly used for implementing the demigration step in least-squares prestack depth migration. In the first part of this chapter the two variants of the focal transform are derived from acoustic WRW modeling of primaries. Deblending will enter the picture again in the last part of this chapter.

---

[1] The name WRW modeling stems from the fact that seismic data is modeled by a composition of propagation and reflection operators, the former usually denoted by some variant of the letter W and the latter by a variant of R. The process is discussed later in the chapter.

3.1 WRW modeling

The WRW modeling that will be discussed here is based on wavefield extrapolation using the Rayleigh II integral [Berkhout, 1982]. The basic concept behind this form of modeling is to separately handle propagation and scattering. This makes it different from other popular modeling strategies that do not decouple scattering and propagation, such as finite difference/finite element methods working directly with the wave equation (see e.g. Kelly et al. [1976]). Decoupling scattering and propagation allows us to parameterize modeling in terms of a reflectivity model (for the scattering part) and a velocity model (for the propagation part).

I assume that a velocity model is available as a priori knowledge and first consider the simplest case, which is modeling a single reflection event. A seismic source, or array of sources, located at the surface $z = z_0$, generates a wavefield, using a known wavelet. This wavefield propagates downwards until it reaches the plane $z = z_m$, which contains some scaterrers. A portion of the wavefield is then scattered and starts propagating upwards. When it reaches the surface it is recorded either by individual receivers, or by receiver arrays. The source then moves to a different location and the same procedure is repeated. Towards the end of this section this simple model will be expanded to generate primary reflection events for more than one depth level.

This modeling scheme could be further extended to include all orders of surface-related and internal multiple scattering [Berkhout, 2014; Davydenko and Verschuur, 2018]. It can also be extended to handle elastic models [Wapenaar, 1984] and anisotropy [Alshuhail et al., 2014], each extension making the modeled wavefield more realistic. These more advanced features can be important for processes such as migration and velocity model inversion, but the simple, primary-only version for acoustic media will suffice for the focal transform. Therefore, such further extensions will not be discussed.

The single-reflector modeling scenario described above is comprised of five steps:

[1] Wavefield generation

[2] Downward extrapolation

[3] Reflection

[4] Upward extrapolation

[5] Wavefield detection/recording

We will now explore each of these steps in more detail.

▪ 3.1.1 Step 1: Wavefield generation

For the wavefield generation step I define the set of source locations $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{n_s}\}$. All these sources lie on the surface, i.e. $\mathbf{s}_i = (x_{si}, y_{si}, z_0)$. For point sources, each $\mathbf{s}_i$ represents a physical source location. For source arrays $\mathbf{s}_i$ is an "effective" location of the source array, usually taken to be its midpoint. The assumption that all sources are located at the surface is a convention, chosen for convenience. We could substitute the surface with a different datum if that were more convenient.

A source (or source array) $\mathbf{s}_i$ generates an incident, downgoing wavefield $s^+(\mathbf{s}'_j, \mathbf{s}_i; \omega)$, its value known at locations $\mathcal{S}' = \{\mathbf{s}'_1, \mathbf{s}'_2, \ldots, \mathbf{s}'_{n'_s}\}$. These lie on a plane $z = z'_0$, which is parallel to, and placed slightly below the surface plane, i.e. $\mathbf{s}'_j = (x_{s'_j}, y_{s'_j}, z'_0)$. No sources should be present on the plane $z = z'_0$ or below, a requirement when using the Rayleigh II integral in the next step. For each frequency component $\omega \in \Omega$ we can store the incident wavefield into frequency slices of a cube $\mathbf{S}^+(z_0)$. Each frequency slice has dimensions $(n'_s \times n_s)$ and has $s^+(\mathbf{s}'_j, \mathbf{s}_i; \omega)$ as its (j, i)-th element.

There are a couple useful remarks regarding the structure of $\mathbf{S}^+(z_0)$. The first is that each column represents a separate incident wavefield generated by a source (array) located at one of the points in $\mathcal{S}$. Each column can be thought of as the source wavefield for a different modeling experiment, in each of which the source is placed at a different location. The second remark is that although in general $\mathbf{S}^+(z_0)$ is a full matrix, there is a particular case of interest where it is sparse. When the source is a dipole point source and $z'_0 \approx z_0$ then $s^+(\mathbf{s}'_j, \mathbf{s}_i; \omega) = 0$ when $j \neq i$. In other words, each column of $\mathbf{S}^+(z_0)$ has a single nonzero element and $\mathbf{S}^+(z_0)$ as a whole, has a diagonal structure (figure 3.1). For arrays of point dipole sources, each column has a number of nonzero elements equal to the number of array elements. For a monopole source the full matrix has nonzero values, with the diagonal elements having the strongest amplitudes.

▪ 3.1.2 Step 2: Downward extrapolation

For the second step we will make use of wavefield extrapolation, which allows us to use known wavefield values to predict wavefield values at locations further away from the source. In our case, the known wavefield values are those of the incident wavefield at $z = z'_0$. The goal is to generate the wavefield at depth level $z = z_m$, with $z_m > z'_0$. Another way to think about the procedure is as a way to generate the wavefield that would have been recorded by virtual receivers at locations in the set $\mathcal{U}_{z_m} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n_u}\}$, with $\mathbf{u}_k = (x_{u_k}, y_{u_k}, z_m)$. In the context of the focal transform, coordinates $\mathcal{U}_{z_m}$ will be referred to as the focal grid points and the term focal grid will be used for the set $\mathcal{U}_{z_m}$.

The extrapolation step can be carried out via means of the one-way version of the

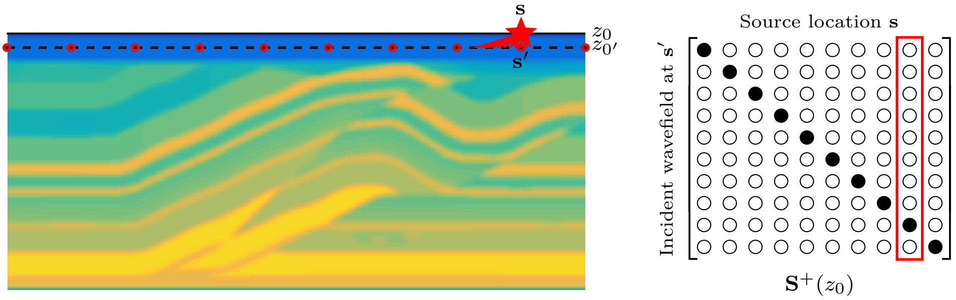


Figure 3.1: *A dipole source at* $\mathbf{s}$ *generates an incident wavefield. The incident wavefield has a nonzero value only at location* $\mathbf{s}'=\mathbf{s}$ *and is stored as the highlighted column of* $\mathbf{S}^+(z_0)$.

Rayleigh II integral, which can be viewed as a specialized variant of the Kirchhoff integral, applicable for extrapolation between planes. The Rayleigh II integral is given by [Wapenaar, 1984, Chapter 5.5.1]

$$p^+(\mathbf{u}_k,\, \mathbf{s}_i;\, \omega) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} 2\frac{\partial G^+(\mathbf{u}_k,\, \mathbf{s}';\, \omega)}{\partial z} s^+(\mathbf{s}',\, \mathbf{s}_i;\, \omega) dx_{s'} dy_{s'}, \tag{3.1.1}$$

where $\mathbf{s}' = (x_{s'}, y_{s'}, z_0')$, $p^+(\mathbf{u}_k,\, \mathbf{s}_i;\, \omega)$ is the downgoing wavefield at a grid point $\mathbf{u}_k$. $G^+(\mathbf{u}_k,\, \mathbf{s}';\, \omega)$ is a one-way Green's function from $\mathbf{s}'$ to $\mathbf{u}_k$. Equation (3.1.1) implies that the incident wavefield should be known everywhere on the infinite plane $z = z_0'$. For practical applications the incident wavefield will be only known at a finite number of discrete locations $\mathcal{S}'$, therefore the integral will have to be numerically approximated and truncated to a finite aperture:

$$\begin{aligned} p^+(\mathbf{u}_k,\, \mathbf{s}_i;\, \omega) \approx \sum_{j=1}^{n_{s'}} 2\frac{\partial G^+(\mathbf{u}_k,\, \mathbf{s}_j';\, \omega)}{\partial z} s^+(\mathbf{s}_j',\, \mathbf{s}_i;\, \omega)\Delta\mathbf{s}_j' = \\ \sum_{j=1}^{n_{s'}} w^+(\mathbf{u}_k,\, \mathbf{s}_j';\, \omega) s^+(\mathbf{s}_j',\, \mathbf{s}_i;\, \omega), \end{aligned} \tag{3.1.2}$$

with

$$w^+(\mathbf{u}_k,\, \mathbf{s}_j';\, \omega) = 2\frac{\partial G^+(\mathbf{u}_k,\, \mathbf{s}_j';\, \omega)}{\partial z}\Delta\mathbf{s}_j'.$$

The term $\Delta\mathbf{s}_j'$ scales $s^+(\mathbf{s}_j',\, \mathbf{s}_i;\, \omega)$ to account for the sampling density around $\mathbf{s}_j'$. Its value will depend on the integration scheme used.

Since the modeling scheme treats propagation and scattering independently, the Green's function used in equation (3.1.2) should not generate any reflection events.

In this sense I refer to it as a one-way Green's function. For homogeneous media this no-reflection condition is automatically fulfilled, as there is no impedance contrast present. In this case, we may say that the two-way and one-way Green's function are identical. The two differ, however, when the medium between z'_0 and z_m is inhomogeneous. Inhomogeneous media will generally have impedance contrasts, giving rise to reflection events and internal multiples related to those events. These events will be present in a true Green's function. Additionally, a true Green's function linking $\mathbf{s}'_j$ and $\mathbf{u}_k$ will contain all orders of surface-related multiples as well.

In order to retain the separation of propagation and scattering, an approximate one-way Green's function will be needed. The requirements are that it should not contain scattering, but should have accurate kinematics. These are constructed using a number of approaches, such as raytracing [Červený and Hron, 1980], Gaussian beams [Červený et al., 1982] and finite-differences [Kelly et al., 1976; Operto et al., 2007] for a smooth version of the inhomogeneous medium. For laterally invariant or smoothly varying media, a convenient option is recursive extrapolation [Berkhout, 1982; Gazdag and Sguazzero, 1984; Thorbecke et al., 2004]. In the special case of a medium varying only along the z-axis, the Green's function between $\mathbf{s}'_j$ and $\mathbf{u}_k$ depends only on their distance, i.e., $w^+(\mathbf{u}_k,\, \mathbf{s}'_j;\, \omega) = w^+(\mathbf{u}_k - \mathbf{s}'_j,\, \mathbf{0};\, \omega)$ [Wapenaar, 1984, Chapter 5.5.1]. Then, equation (3.1.2) becomes a spatial convolution in the frequency-space domain, or element-by-element multiplication with a phase-shift operator in the frequency-wavenumber domain [Berkhout, 1982].

For media with homogeneous velocity, analytic expressions can be derived for $w^+(\mathbf{u}_k,\, \mathbf{s}'_j;\, \omega)$. Table 3.1 lists formulae for two- and three-dimensional media. The derivations of these expressions can be found in Gisolf and Verschuur [2010].

Space	3D dipole	2D dipole
Exact solution	$\frac{z_m(1+j\omega\tau)}{\Delta r^3} e^{-j\omega\tau}$	$-j\omega\pi \frac{z_m}{c\Delta r} H_1^{(2)}(\omega\tau)$
Far field approximation	$j\omega \frac{z_m}{c\Delta r^2} e^{-j\omega\tau}$	$z_m \sqrt{\frac{2\pi j\omega}{\Delta r^3}} e^{-j\omega\tau}$
	$\Delta r = \lVert\mathbf{s}'_j - \mathbf{u}_k\rVert_2, \quad \tau = \Delta r / c$	

Table 3.1: *Analytic expressions for the vertical derivative of the Green's function from $\mathbf{s}'_j$ to $\mathbf{u}_k$ of a medium with homogeneous velocity c. Here $H_1^{(2)}(\cdot)$ is a first order Hankel function of the second kind.*

For each frequency ω we can construct a frequency slice of $\mathbf{W}^+_m(z_m,\, z_0)$, by placing $w^+(\mathbf{u}_k,\, \mathbf{s}'_j;\, \omega)$ as the (k, j)-th element of the slice. Each slice then has dimensions $(n_u \times n'_s)$. The downgoing wavefield that is generated by the extrapolation for all

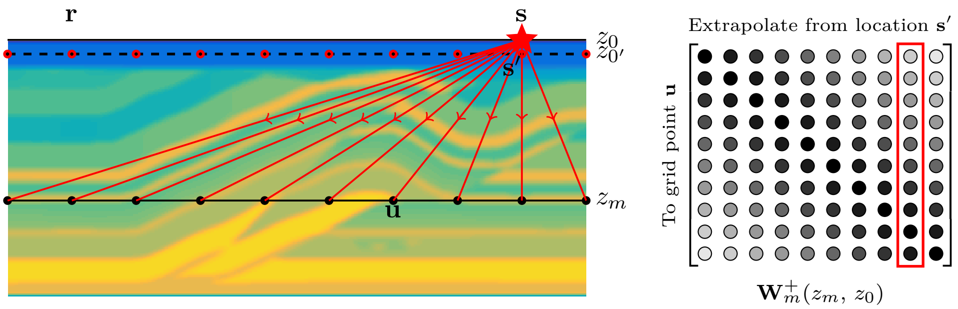


Figure 3.2: *The downgoing wavefield is extrapolated from* z_0' *to* z_m*. This is carried out by* $\mathbf{W}_m^+(z_m,\, z_0)$*. Shown here is the contribution of the highlighted column of* $\mathbf{W}_m^+(z_m,\, z_0)$*. The straight red lines demonstrate extrapolation schematically and are not meant to represent accurate raypaths.*

sources can be also placed into a $(n_u \times n_s)$ data cube $\mathbf{P}^+(z_m,\, z_0)$, using $p^+(\mathbf{u}_k,\, \mathbf{s}_i;\, \omega)$ as the (k, i)-th element. The two are linked by

$$\mathbf{P}^+(z_m,\, z_0) = \mathbf{W}_m^+(z_m,\, z_0)\mathbf{S}^+(z_0). \tag{3.1.3}$$

Each column of $\mathbf{P}^+(z_m,\, z_0)$ contains the downgoing wavefield generated by sources at different locations. A depiction of the extrapolation process can be seen in figure 3.2. For the purposes of focal deblending as described later on I approximate the kinematics of propagation between z_0 and z_m using a homogeneous velocity model and construct $\mathbf{W}_m^+(z_m,\, z_0)$ using the exact analytic expressions found in table 3.1.

■ 3.1.3 Step 3: Reflection

Up to this point the wavefield has been traveling from z_0 to z_m without encountering any scattering, and therefore it has been purely downgoing. In the third step, this changes as the wavefield now encounters scatterers present on the $z = z_m$ plane. The scatterers partially convert the downgoing wavefield into a reflected, upgoing wavefield (figure 3.3). The amplitude of the reflected wavefield at a scatterer location is determined by the reflectivity of that scatterer, which is a function of the acoustic impedance contrast between the scatterer and the background medium, as well as the angle of incidence of the downgoing wavefield [Berkhout, 1982; de Bruin et al., 1990]. The downgoing wavefield can be thought of as a superposition of plane waves, each of which has a well defined angle of incidence with respect to the normal at the scatterer location. Each of these plane waves is then scaled by a different reflectivity coefficient, depending on its angle. The superposition of the reflected plane waves is the total upgoing wavefield. Here the combination of plane-wave decomposition,

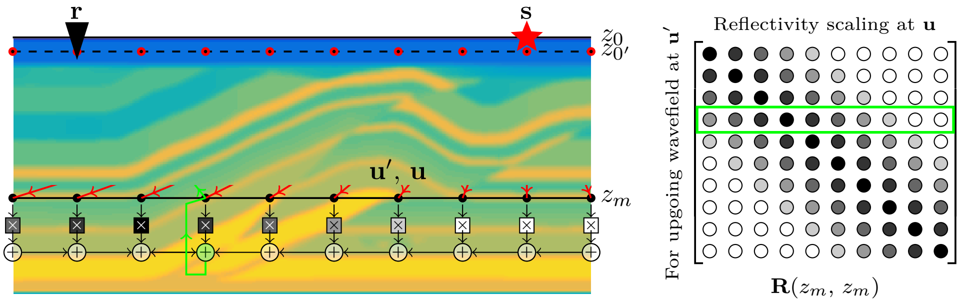


Figure 3.3: *The downgoing wavefield at* z_m *(red arrows), is converted into an upgoing wavefield (green arrow for the fourth focal grid point) by* $\mathbf{R}(z_m,\, z_m)$*. The operation is nonlocal; each upgoing field sample is given as an inner product of the downgoing wavefield at each focal grid point with a reflection operator (here corresponding to the fourth row of* $\mathbf{R}(z_m,\, z_m)$*). The off-diagonal elements of* $\mathbf{R}(z_m,\, z_m)$ *give rise to angle-dependent reflectivity.*

angle-based scaling, and plane-wave superposition is carried out as one single step in the space domain by means of a non-stationary convolution.

In order to capture the angle-dependent nature of the reflectivity, each scatterer at grid point $\mathbf{u}'_l \in \mathcal{U}_{z_m}$ is replaced by an effective group of point scatterers, that is given by a subset of $\mathcal{U}_{z_m}$ that is spatially close to $\mathbf{u}'_l$. For each pair $(\mathbf{u}'_l, \mathbf{u}_k)$ and frequency ω we define a reflectivity coefficient[2] $r(\mathbf{u}'_l,\, \mathbf{u}_k;\, \omega)$ with $r(\mathbf{u}'_l,\, \mathbf{u}_k;\, \omega) = 0$ for those grid points $\mathbf{u}_k$ that do not contribute in explaining the angle-dependent reflectivity at $\mathbf{u}'_l$. Then,

$$p^-(\mathbf{u}'_l,\, \mathbf{s}_i;\, \omega) = \sum_{k=1}^{n_u} r(\mathbf{u}'_l,\, \mathbf{u}_k;\, \omega) p^+(\mathbf{u}_k,\, \mathbf{s}_i;\, \omega). \tag{3.1.4}$$

Note that in a laterally homogeneous medium the operators $r(\mathbf{u}'_l,\, \mathbf{u}_k;\, \omega)$ only depend on the distance $||\mathbf{u}'_l - \mathbf{u}_k||_2$ due to translational invariance. This is no longer the case for laterally inhomogeneous media, however. As before, we place $p^-(\mathbf{u}'_l,\, \mathbf{s}_i;\, \omega)$ and $r(\mathbf{u}'_l,\, \mathbf{u}_k;\, \omega)$ in cubes $\mathbf{P}^-(z_m,\, z_0)$ and $\mathbf{R}(z_m,\, z_m)$, respectively. Each monochromatic frequency slice of $\mathbf{P}^-(z_m,\, z_0)$ has dimensions $(n_u \times n_s)$ and each slice of $\mathbf{R}(z_m,\, z_m)$ $(n_u \times n_u)$. This yields

$$\mathbf{P}^-(z_m,\, z_0) = \mathbf{R}(z_m,\, z_m)\mathbf{P}^+(z_m,\, z_0). \tag{3.1.5}$$

For the special case of locally reacting media, $r(\mathbf{u}'_l,\, \mathbf{u}_k;\, \omega) = 0$ for $\mathbf{u}'_l \neq \mathbf{u}_k$, each frequency slice of $\mathbf{R}(z_m,\, z_m)$ becomes a diagonal matrix. For focal subdomains,

---

[2] While possible to derive from medium parameters, in practice the values $r(\mathbf{u}'_l,\, \mathbf{u}_k;\, \omega)$ would be estimated by migration, as the WRW modeling described here serves primarily as a demigration process.

which play a role analogous to $\mathbf{R}(z_m,\, z_m)$, no specific structure is imposed and in principle every element could have a nonzero value.

■ 3.1.4 Step 4: Upward extrapolation

The reflected portion of the wavefield starts traveling upwards until it reaches the receivers at the surface. Generating the wavefield at $z = z_0$ knowing its values at $z = z_m$, is a second forward extrapolation problem, similar to that of step 2. The difference is that now the scatterers act as virtual sources and the receivers are now located at the surface, away from the scatterers. For this extrapolation step we need one-way Green's functions from $\mathbf{u}'_l$ to point-receivers $\mathbf{r}'_o$. The discretized Rayleigh II integral for this scenario then is given by

$$\begin{aligned} p^-(\mathbf{r}'_o,\, \mathbf{s}_i;\, \omega) \approx & \sum_{l=1}^{n_u} 2\frac{\partial G^-(\mathbf{r}'_o,\, \mathbf{u}_l;\, \omega)}{\partial z} s^+(\mathbf{u}_l,\, \mathbf{s}_i;\, \omega)\Delta\mathbf{u}_l = \\ & \sum_{l=1}^{n_u} w^-(\mathbf{r}'_o,\, \mathbf{u}_l;\, \omega) s^+(\mathbf{s}'_j,\, \mathbf{s}_i;\, \omega), \end{aligned} \tag{3.1.6}$$

with

$$w^-(\mathbf{r}'_o,\, \mathbf{u}_l;\, \omega) = 2\frac{\partial G^-(\mathbf{r}'_o,\, \mathbf{u}'_l;\, \omega)}{\partial z}\Delta\mathbf{u}'_l. \tag{3.1.7}$$

After placing all values $p^-(\mathbf{r}'_o,\, \mathbf{s}_i;\, \omega)$ in the data cube $\mathbf{P}^-(z_0,\, z_0)$ and $w^-(\mathbf{r}'_o,\, \mathbf{u}_l;\, \omega)$ in $\mathbf{W}^-_m(z_0,\, z_m)$ enables us to rewrite (3.1.6) as

$$\mathbf{P}^-(z_0,\, z_0) = \mathbf{W}^-_m(z_0,\, z_m)\mathbf{P}^-(z_m,\, z_0). \tag{3.1.8}$$

For homogeneous media $w^-(\mathbf{x}_1,\, \mathbf{x}_2;\, \omega) = w^+(\mathbf{x}_2,\, \mathbf{x}_1;\, \omega)$ for two points $\mathbf{x}_1$, $\mathbf{x}_2$ and, therefore, $\mathbf{W}^-_m(z_0,\, z_m) = [\mathbf{W}^+_0(z_0,\, z_m)]^{\mathrm{T}}$.

■ 3.1.5 Step 5: Wavefield detection/recording

In the previous step the reflected wavefield was extrapolated to the surface at locations of the individual receivers. Often that is the end of modeling, however in some cases it might be desirable to also emulate the effect of receiver arrays on the data and/or apply receiver frequency responses. Receiver arrays can be easily incorporated in the modeling scheme by taking linear combinations of the recordings of groups of receivers, i.e.,

$$p(\mathbf{r}_q,\, \mathbf{s}_i;\, \omega) = \sum_{o=1}^{n_r} d(\mathbf{r}_q,\, \mathbf{r}'_o;\, \omega) p^-(\mathbf{r}'_o,\, \mathbf{s}_i;\, \omega). \tag{3.1.9}$$

Here $\mathbf{r}_q \in \mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{n_r}\}$ is the effective location of the array and $\mathbf{r}'_o$ the location of an individual receiver. For receivers $\mathbf{r}'_o$ not part of the array at $\mathbf{r}_q$, $d(\mathbf{r}_q,$

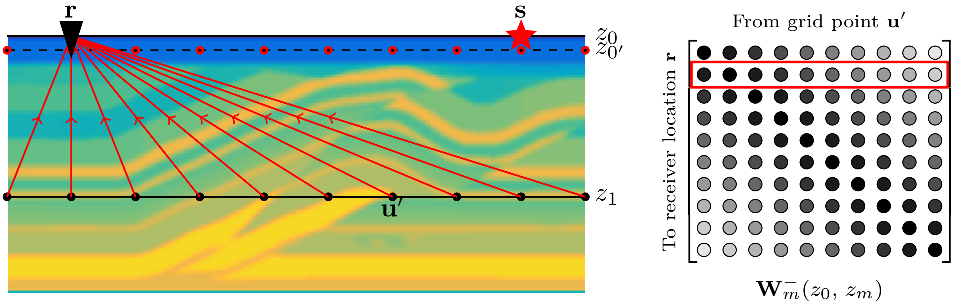


Figure 3.4: *The upgoing wavefield at z_m is extrapolated to z_0. Schematically depicted here is the contribution of the highlighted row of $\mathbf{W}_m^-(z_0,\, z_m)$. The straight red lines demonstrate extrapolation schematically and are not meant to represent accurate raypaths.*

$\mathbf{r}_o'; \omega) = 0$. For those that are part of the array, $d(\mathbf{r}_q,\, \mathbf{r}_o';\, \omega)$ takes the form of a complex value that is the frequency response at ω of the receiver placed at $\mathbf{r}_o'$. When a flat frequency response is desired, usually $d(\mathbf{r}_q,\, \mathbf{r}_o';\, \omega) = 1$. Using the data cube notation,

$$\mathbf{P}(z_0,\, z_0) = \mathbf{D}(z_0)\mathbf{P}^-(z_0,\, z_0), \tag{3.1.10}$$

where $\mathbf{P}(z_0,\, z_0)$ and $\mathbf{D}(z_0)$ are formed from $p(\mathbf{r}_q,\, \mathbf{s}_i;\, \omega)$ and $d(\mathbf{r}_q,\, \mathbf{r}_o';\, \omega)$, in a manner analogous to the previous steps. When no array-forming takes place, $\mathbf{D}(z_0)$ has one nonzero element per row.

■ 3.1.6 Multi-level modeling and migration

Combining equations (3.1.3), (3.1.5), (3.1.6) and (3.1.10) we arrive at

$$\mathbf{P}(z_0,\, z_0) = \mathbf{D}(z_0)[\mathbf{W}_m^-(z_0,\, z_m)\mathbf{R}(z_m,\, z_m)\mathbf{W}_m^+(z_m,\, z_0)]\mathbf{S}^+(z_0), \tag{3.1.11}$$

which combines all the modeling steps discussed previously. Modeling primary reflections from $M > 1$ depth levels can be done by repeating these steps and summing the reflection data produced at each depth level. Let $z_m = m\Delta z$ with $m = 1, 2, \ldots, M$. The total wavefield at the surface then becomes,

$$\mathbf{P}(z_0,\, z_0) = \mathbf{D}(z_0)\sum_{m=1}^{M}[\mathbf{W}_m^-(z_0,\, z_m)\mathbf{R}(z_m,\, z_m)\mathbf{W}_m^+(z_m,\, z_0)]\mathbf{S}^+(z_0). \tag{3.1.12}$$

Transmission effects are neglected here. Here 3.1.12 is used in the forward sense, mapping $\mathbf{R}(z_m,\, z_m)$, $m = 1, 2, \ldots, M$ to $\mathbf{P}(z_0,\, z_0)$, assuming a known velocity model. The inverse process assumes that observed data $\mathbf{P}_{\text{obs}}(z_0,\, z_0)$ and the velocity model are given and aims to recover $\mathbf{R}(z_m,\, z_m)$, $m = 1, 2, \ldots, M$. This process is commonly referred to as migration. Often the additional requirement of consistency with the observed data is placed on migration. Perhaps the most common way to enforce consistency is to find $\mathbf{R}_{\text{migr}}(z_m, z_m)$, $m = 1, 2, \ldots, M$, that minimize the ℓ_2-distance of the modeled and observed data, yielding a form of least-squares migration:

$$\mathbf{R}_{\text{migr}}(z_m, z_m) = \underset{\substack{\mathbf{R}(z_m, z_m),\\ m=1,2,\ldots,M}}{\arg\min} \{\sum_{\omega} ||\mathbf{P}_{\text{obs}}(z_0, z_0) - \mathbf{D}(z_0) \sum_{m=1}^{M} [\mathbf{W}_m^-(z_0, z_m)\mathbf{R}(z_m, z_m)\mathbf{W}_m^+(z_m, z_0)]\mathbf{S}^+(z_0)||_{\text{F}}\}. \quad (3.1.13)$$

It is possible to place additional constraints on the structure of each $\mathbf{R}(z_m, z_m)$, depending on whether the recovered reflectivity is to be angle-dependent. Figure 3.5 shows a subsurface image after migrating 20 shot gathers of synthetic data using inverse extrapolation. The m-th row of the image is given by the diagonal of $\hat{\mathbf{R}}(z_m, z_m)$ at $t = 0$ s. Note that Berkhout [2014] describes a process in which, using the same building blocks, multiple scattering can also be included. However, for our purpose this is not relevant. In the context of focal deblending multiples will generally be explained as fictitious primary events.

3.2 Inverse extrapolation and focusing

The WRW modeling engine described in the previous section uses two forward extrapolation steps (figure 3.6, left). The first extrapolates the wavefield further away from the source, towards the interior of the medium. The second extrapolates the reflected, upgoing wavefield away from the reflector.

At the heart of migration, redatuming and the focal transform, is the ability to carry out the inverse process, known as *inverse extrapolation.* This operation extrapolates a wavefield closer to its source (figure 3.6, right). In the ideal case inverse extrapolation would consist of applying operators $\mathbf{W}_i^+(z_i, z_0)^{-1}$ and $\mathbf{W}_i^-(z_0, z_i)^{-1}$. A well-documented problem of this approach is that it exponentially amplifies the evanescent part of the wavefield, attempting to reverse the exponential decay it experiences during forward extrapolation. The recorded wavefield typically has a very low signal-to-noise ratio in the evanescent part of the frequency-wavenumber spectrum, as it is recorded many wavelengths away from the reflectors. The amplification then, ends up amplifying mostly noise. Additionally, $\mathbf{W}_i^+(z_i, z_0)$ and $\mathbf{W}_i^-(z_0, z_i)$ are badly-conditioned, making the process of constructing their inverses instable [Berkhout, 1982; Wapenaar, 1984].

A practical approach that overcomes these difficulties of inverse extrapolation is to have the inverse extrapolation operators take the form of the Hermitian of the corresponding forward propagation operators, i.e. $\mathbf{W}_i^+(z_i, z_0)^{\text{H}}$ and $\mathbf{W}_i^-(z_0, z_i)^{\text{H}}$. These

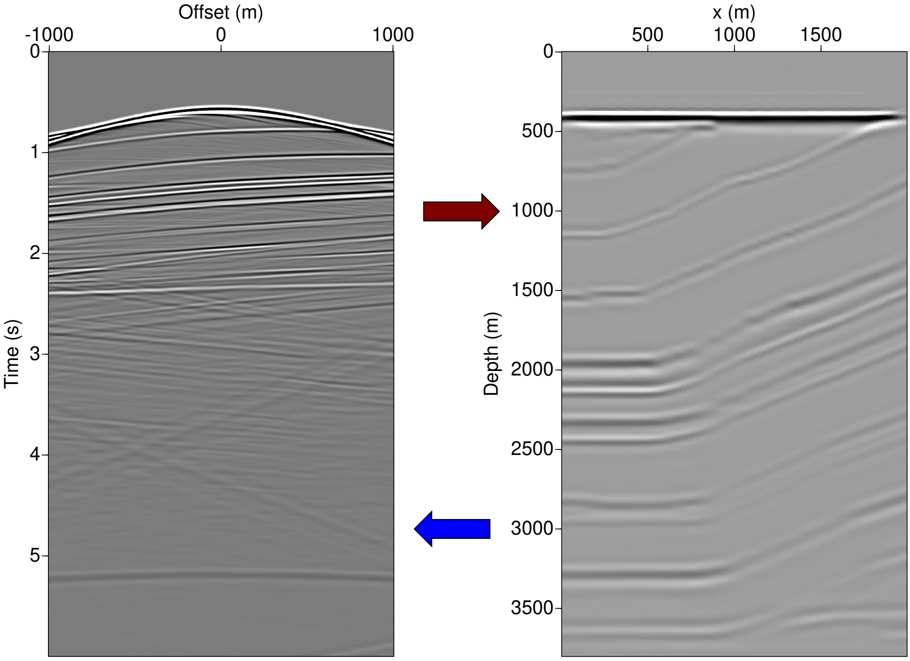


Figure 3.5: *Left: a synthetic, prestack shot gather. Right: the image after migration. The migration process (red arrow) maps a prestack data cube to a reflectivity image. Demigration, e.g. WRW modeling, maps a reflectivity image to surface data.*

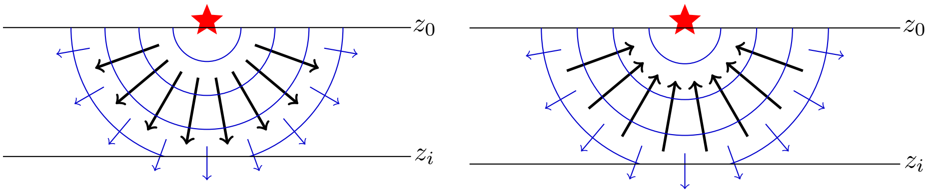


Figure 3.6: *Left: forward extrapolation. Right: inverse extrapolation. Blue arrows denote the direction of the wavefield propagation and black arrows the direction of the extrapolation.*

operators remove propagation effects, without attempting to recover the evanescent wavefield. When applied to $\mathbf{P}(z_0, z_0)$, they remove transient propagation effects up to depth level z_i. Provided that the operators are kinematically accurate, this action *focuses* that component of the dataset which is the result of reflection that occured at depth level z_i. This focusing is visible in the time/subsurface-offset domain. This action brings the entire reflector into focus, provided that one exists at z_i and that the velocity model is sufficiently accurate. As was the case for the forward extrapolation operators, these inverse extrapolation operators are scattering-free.

Given a dataset $\mathbf{P}(z_0,\, z_0)$, focusing at z_i takes the form

$$\begin{aligned}
\boldsymbol{\mathcal{X}}_i^D(z_i,\, z_i) &= \mathbf{W}_i^-(z_0,\, z_i)^{\mathrm{H}}\, \mathbf{P}(z_0,\, z_0)\, \mathbf{W}_i^+(z_i,\, z_0)^{\mathrm{H}} \\
&= \mathbf{W}_i^-(z_0,\, z_i)^{\mathrm{H}} \Big[\mathbf{D}(z_0) \sum_{m=1}^{M} \mathbf{W}_m^-(z_0,\, z_m)\mathbf{R}(z_m,\, z_m)\times \\
&\qquad\qquad \times \mathbf{W}_m^+(z_m,\, z_0)\mathbf{S}^+(z_0)\Big] \mathbf{W}_i^+(z_i,\, z_0)^{\mathrm{H}} \\
&= \mathbf{W}_i^-(z_0,\, z_i)^{\mathrm{H}} \Big[\mathbf{I} \sum_{m=1}^{M} \mathbf{W}_m^-(z_0,\, z_m)\mathbf{R}(z_m,\, z_m)\times \\
&\qquad\qquad \times \mathbf{W}_m^+(z_m,\, z_0)\big(S(\omega)\mathbf{I}\big)\Big] \mathbf{W}_i^+(z_i,\, z_0)^{\mathrm{H}} \\
&= \mathbf{W}_i^-(z_0,\, z_i)^{\mathrm{H}} \Big[\sum_{m=1}^{M} \mathbf{W}_m^-(z_0,\, z_m)\big(S(\omega)\mathbf{R}(z_m,\, z_m)\big)\times \\
&\qquad\qquad \times \mathbf{W}_m^+(z_m,\, z_0)\Big] \mathbf{W}_i^+(z_i,\, z_0)^{\mathrm{H}}.
\end{aligned}$$

Note that here I have assumed that $\mathbf{S}^+(z_0) = S(\omega)\mathbf{I}$ and $\mathbf{D}(z_0) = \mathbf{I}$, i.e. no source/receiver arrays and the same source signature $S(\omega)$ for all shots. These assumptions will be also part of the definition of the focal transform later on. In particular, source wavelets are not assumed to be known and an average wavelet will be imprinted onto the focal subdomains to compensate. Moving $\mathbf{W}_i^-(z_0,\, z_i)^{\mathrm{H}}$ and $\mathbf{W}_i^+(z_i,\, z_0)^{\mathrm{H}}$ inside the sum yields

$$\begin{aligned}
\boldsymbol{\mathcal{X}}_i^D(z_i,\, z_i) &= \sum_{m=1}^{M} \left[\mathbf{W}_i^-(z_0,\, z_i)^{\mathrm{H}}\mathbf{W}_m^-(z_0,\, z_m)\right] \left[S(\omega)\mathbf{R}(z_m,\, z_m)\right]\times \\
&\qquad\qquad \times \left[\mathbf{W}_m^+(z_m,\, z_0)\mathbf{W}_i^+(z_i,\, z_0)^{\mathrm{H}}\right] \\
&= \sum_{m=1}^{M} \mathbf{Q}^-(z_i,\, z_m)\big[S(\omega)\mathbf{R}(z_m,\, z_m)\big]\mathbf{Q}^+(z_m,\, z_i), \qquad (3.2.14)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{Q}^-(z_i,\, z_m) &= \mathbf{W}_i^-(z_0,\, z_i)^{\mathrm{H}}\mathbf{W}_m^-(z_0,\, z_m), \\
\mathbf{Q}^+(z_m,\, z_i) &= \mathbf{W}_m^+(z_m,\, z_0)\mathbf{W}_i^+(z_i,\, z_0)^{\mathrm{H}}.
\end{aligned}$$

Equation (3.2.14) can be split into a contribution from the reflector at z_i and a contribution from the rest of the reflectors,

$$
\begin{aligned}
\boldsymbol{\mathcal{X}}_i^D(z_i,\, z_i) &= \sum_{m=1}^{M} \mathbf{Q}^-(z_i,\, z_m)\big[S(\omega)\mathbf{R}(z_m,\, z_m)\big]\mathbf{Q}^+(z_m,\, z_i) \\
&= \underbrace{\mathbf{Q}^-(z_i,\, z_i)\big[S(\omega)\mathbf{R}(z_m,\, z_m)\big]\mathbf{Q}^+(z_i,\, z_i)}_{\text{Focused event}} + \\
&\qquad + \sum_{\substack{m=1 \\ m\neq i}}^{M} \underbrace{\mathbf{Q}^-(z_i,\, z_m)\big[S(\omega)\mathbf{R}(z_m,\, z_m)\big]\mathbf{Q}^+(z_m,\, z_i)}_{\text{Unfocused events}} .
\end{aligned}
$$

An example slice of $\boldsymbol{\mathcal{X}}_i^D(z_i,\, z_i)$ corresponding to one lateral position x is shown in figure 3.7, where we consider the response of a 5-reflector medium. We can clearly see the difference in character between the two contributing components. The dataset shown in figure 3.7 was focused at z_3. It was modeled using the WRW modeling described previously, and accurate extrapolation operators were used for the focusing process. Unlike the other four events, the reflection from z_3 has been focused at time zero and offset zero. The time slice at time zero can be thought of as a migrated image sampled at depth z_i.

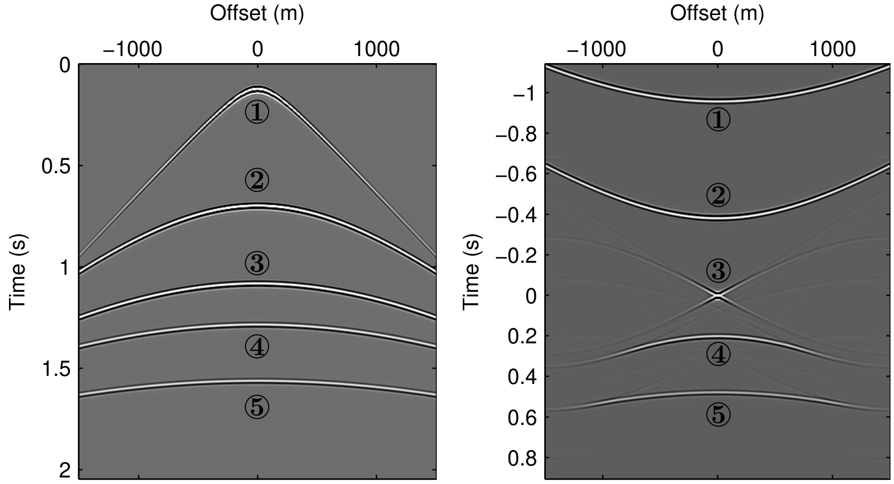


Figure 3.7: *Left: a gather with five events, modeled using WRW modeling. Right: a gather from the same dataset, focused at the reflector producing the third primary event. The numbers match events with their counterparts in the focused gather.*

Several practical limitations prevent the focusing operation from exposing the true reflectivity operator. The first is that, due to the lack of the evanescent part, $\mathbf{Q}^-(z_i,\, z_i) \neq \mathbf{I}$ and $\mathbf{Q}^+(z_i,\, z_i) \neq \mathbf{I}$. The combined effect of $\mathbf{Q}^-(z_i,\, z_i)$ and $\mathbf{Q}^+(z_i,\, z_i)$ acts as a point spread function that distorts the reflectivity operator[3]. Together with the presence of $S(\omega)$, the point spread function places an upper limit on the spatial and temporal resolution of $\boldsymbol{\mathcal{X}}_i^D(z_i,\, z_i)$. An additional limitation is the finite aperture of the acquisition. For field data, aliasing often is an issue, as sampling is rarely fine enough in all dimensions. Despite these shortcomings, it is clear from figure 3.7 that the primary event has a significant reduction in spatial and temporal extent when focused. We will make use of this property for deblending later on.

3.3 The multi-level double focal transform

Building on the material of the previous chapter we can now define the double focal transform. The basis for defining the transform is WRW modeling, with a few adjustments:

- The focal transform uses K depth levels, z_k, $k = 1, 2, \ldots, K$, with $K \ll M$. Unlike in migration, these are generally few, not equispaced and $z_{k+1} - z_k$ is larger than what would be typically used in a migration scheme. It is also possible to use non-flat levels $z_k(x, y)$.

- The reflectivity cube is replaced by a *focal subdomain* $\mathbf{X}_k^D(z_k,\, z_k)$, asssociated with depth z_k. The important difference is that, unlike a reflectivity cube, a focal subdomain is not restricted to have any other particular structure. $\mathbf{X}_k^D(z_k,\, z_k)$ differs from $\boldsymbol{\mathcal{X}}_k^D(z_k,\, z_k)$ introduced earlier. The latter is the result of applying a single inverse extrapolation step, whereas the former is used for the results of inversion, or when synthesizing surface data from focal subdomains.

- The extrapolation operators $\mathbf{W}_k^-(z_0,\, z_k)$ and $\mathbf{W}_k^+(z_k,\, z_0)$ are constructed assuming a homogeneous, isotropic acoustic medium with acoustic velocity c_k. This velocity can be thought of as an average velocity of the subsurface layers between z_0 and z_k. I will refer to these as *focal* operators. These velocities are considered prior knowledge. It is possible to use an inhomogeneous velocity model as well, but in any case the velocity model is not assumed to be accurate. If a layered velocity model is available, then more accurate operators $\mathbf{W}_k^+(z_k,\, z_0)$ and $\mathbf{W}_k^-(z_0,\, z_k)$ can be constructed by composing operators for each homogeneous layer, e.g. $\mathbf{W}_k^+(z_k,\, z_0) = \prod_{i=1}^{k} \mathbf{W}^+(z_i,\, z_{i-1})$.

- For the double focal transform discussed here, $\mathbf{D}(z_0) = \mathbf{I}$ and $\mathbf{S}^+(z_0) = \mathbf{I}$, i.e. we do not assume prior knowledge of the source and receiver characteristics.

---

[3] An analytic expression for homogeneous acoustic media is derived in Wapenaar [1984].

These adjustments come from practical considerations. The main idea behind the focal transform is to have a data preprocessing tool that leverages, to some extent, the physics of wave propagation. At the first stages of processing information such as detailed velocity models and wavelet shapes are generally not available, so I opt not to use such information.

On the other hand, when such extra information is available it can be incorporated in the formulation of the transform. This can happen, for example, in the case of a 4D monitor survey, where much information will be available from processing the base survey. At the limit, when using the exact velocity model and densely sampled depth levels, the focal transform becomes a migration/modeling algorithm that has been repurposed.

■ 3.3.1 The inverse transform

Applying the above adjustments to (3.1.12) gives us the *inverse* focal transform:

$$\mathbf{P}(z_0, z_0) = \sum_{k=1}^{K} \mathbf{W}_k^-(z_0, z_k)\mathbf{X}_k^D(z_k, z_k)\mathbf{W}_k^+(z_k, z_0). \tag{3.3.15}$$

Since (3.3.15) takes us from the transform domain, that is the content of $\mathbf{X}_k^D(z_k, z_k)$, $k = 1, 2, \ldots, K$, to the data domain, which here is $\mathbf{P}(z_0, z_0)$, I define it as the inverse transform, in analogy to other transforms, such as Fourier, wavelet, etc. Note that (3.3.15) is a simplified version of forward modeling.

■ 3.3.2 The forward transform

Calculating the forward focal transform presents a difficulty, when focusing on multiple depth levels. Focal subdomains usually have a size comparable to the original data. When multiple subdomains are used, the forward focal transform tends to become an underdetermined problem. For the purposes of this thesis I define the forward focal transform as the following optimization problem:

$$\left\{\hat{\mathbf{X}}_{\text{rec},1}^D(z_1, z_1), \hat{\mathbf{X}}_{\text{rec},2}^D(z_2, z_2), \ldots, \hat{\mathbf{X}}_{\text{rec},K}^D(z_K, z_K)\right\} =$$

$$\underset{\substack{\hat{\mathbf{X}}_k^D(z_k, z_k)\\ k=1,\ldots,K}}{\arg\min} \left\{\sum_t \sum_{k=1}^{K} \left|\left|\hat{\mathbf{X}}_k^D(z_k, z_k)\right|\right|_{\text{S}}\right\} \quad \text{subject to}$$

$$\sum_\omega \left|\left|\mathbf{P}(z_0, z_0) - \sum_{k=1}^{K} \mathbf{W}_k^-(z_0, z_k)\mathbf{X}_k^D(z_k, z_k)\mathbf{W}_k^+(z_k, z_0)\right|\right|_{\text{F}} \leq \sigma. \tag{3.3.16}$$

Note that the hat diacritic means that $\hat{\mathbf{X}}_{\text{rec},k}^D(z_k, z_k)$ is in the time domain. The optimization problem (3.3.16) yields focal subdomains $\hat{\mathbf{X}}_{\text{rec},k}^D(z_k, z_k)$ with the smallest total sum norm, such that the total reconstruction misfit is bounded by σ. The total reconstruction misfit is the constraint in (3.3.16) and uses the Frobenius norm as a measure of distance.

It should be mentioned that defining the forward focal transform as in (3.3.16) is not the only possibility. In fact, any definition that fulfills the following criterion could be a candidate:

$$\operatorname{dist}\left(\mathbf{P}(z_0,\, z_0),\, \sum_{k=1}^{K} \mathbf{W}_k^-(z_0,\, z_k)\mathbf{X}_k^D(z_k,\, z_k)\mathbf{W}_k^+(z_k,\, z_0)\right) \leq \mu < \infty, \tag{3.3.17}$$

for some choice of distance function $\operatorname{dist}(\cdot,\, \cdot)$. Any number of additional constraints may be placed on $\mathbf{X}_k^D(z_k,\, z_k)$. The choice of the distance function, the value of μ and additional constraints on the solution space are influenced by the application for which the focal transform is to be used. Practical considerations, such as the availability and computational cost of numerical solvers for the optimization problem may also play a role in the decision.

The formulation (3.3.16) poses the forward focal transform as a basis pursuit denoising (BPDN) problem. It can be shown that after vectorization, (3.3.16) can be rewritten in the form

$$\mathbf{x}_{\text{rec}} = \arg\min_{\mathbf{x}} \{||\mathbf{x}||_1\} \quad \text{subject to} \quad ||\mathbf{p} - \mathbf{\Phi}\mathbf{x}||_2 \leq \sigma, \tag{3.3.18}$$

which is an alternative formulation of the BPDN problem[4] [van den Berg and Friedlander, 2008]. Here $\mathbf{p}$ and $\mathbf{x}$ are vectorizations of the data and focal subdomains, respectively, in the time domain. $\mathbf{\Phi}$ is a time-domain implementation of the inverse focal transform. The proof can be found in Appendix A.

Basis pursuit denoising is a popular tool for signal processing applications [Chen and Donoho, 1998; Sardy et al., 2001; Zhang et al., 2015]. Given measured data $\mathbf{p}$ that is in some way non-ideal, e.g. corrupted by noise, undersampled etc. BPDN produces an estimate

$$\mathbf{p}_{\text{rec}} = \mathbf{\Phi}\mathbf{x}_{\text{rec}}, \tag{3.3.19}$$

that approximates the ideal version of the data,

$$\mathbf{p}_{\text{ideal}} = \mathbf{\Phi}\mathbf{x}_{\text{ideal}}. \tag{3.3.20}$$

The key assumption that BPDN makes is that for the chosen $\mathbf{\Phi}$

$$||\mathbf{x}_{\text{ideal}}||_1 < ||\mathbf{x}||_1\,, \qquad \forall\, \mathbf{x} \neq \mathbf{x}_{\text{ideal}}, \tag{3.3.21}$$

i.e. the ideal solution will have the smallest ℓ_1-norm. The ℓ_1-norm is used here as a

---

[4] The BPDN problem was introduced by Chen et al. [2001] in the form $\mathbf{x}_{\text{opt}} = \arg\min_{\mathbf{x}} \left\{||\mathbf{p} - \mathbf{\Phi}\mathbf{x}||_2 + \lambda\, ||\mathbf{x}||_1\right\}$. Here, I follow van den Berg and Friedlander [2008] and define BPDN as in (3.3.18).

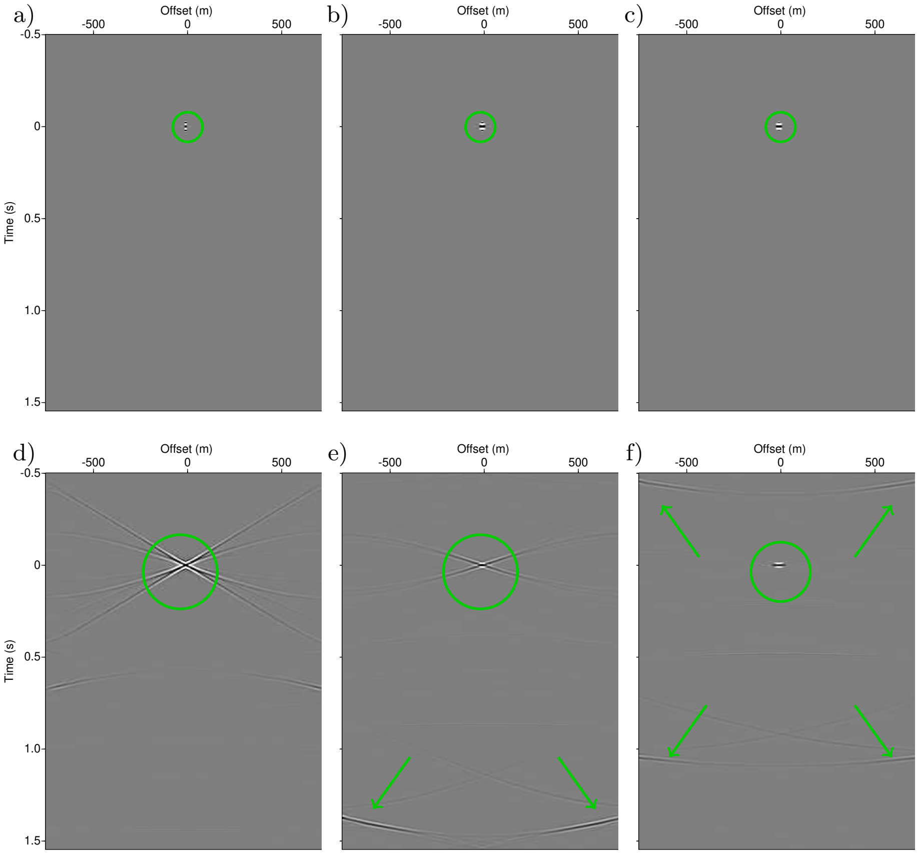


Figure 3.8: *Top row: focal subdomains when minimizing the sum-norm, a) first, b) second, c) third focal subdomain. Bottom row: focal subdomains when minimizing the Frobenius-norm, d) first, e) second, f) third focal subdomain. Using the sum-norm leads to a sparser representation in the focal domain.*

convex approximation to the ℓ_0-pseudonorm[5] [Donoho, 2006]. Both have the effect of promoting sparse solutions [Foucart and Rauhut, 2017, Chapter 3.1], but solving

$$\min_{\mathbf{x}} \{||\mathbf{x}||_0\} \quad \text{subject to} \quad ||\mathbf{p} - \mathbf{\Phi x}||_2 \leq \sigma, \tag{3.3.22}$$

is known to be an NP-hard, combinatorial problem [Donoho, 2006], hence it is impractical for problems where $\mathbf{x}$ has many elements.

---

[5] The ℓ_0-pseudonorm counts the nonzero elements of vector. Strictly speaking, it cannot be called a norm, since $||a\mathbf{x}||_0 \neq |a|\,||\mathbf{x}||_0$, when $a \neq 0$, i.e. the absolute homogeneity property of a norm is violated.

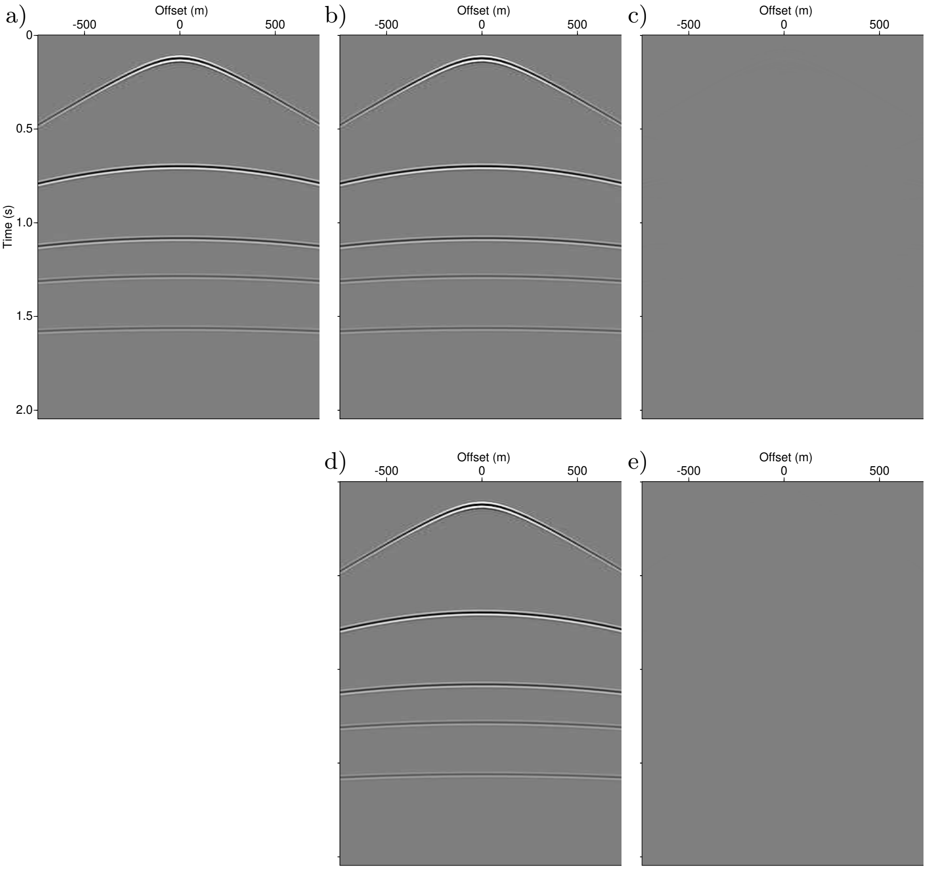


Figure 3.9: *Top row: a) original data, b) reconstructed data by minimizing the sum-norm, c) difference between a) and b). Lower row: d) reconstructed data by minimizing the Frobenius-norm, e) difference between a) and d). The original data is explained in both cases.*

Sparsity gives us a different way to interpret (3.3.21). Loosely speaking, it implies that $\mathbf{p}_{\text{ideal}}$ can be composed by scaling and summing a few columns of $\mathbf{\Phi}$, unlike corrupted versions $\mathbf{p}$, which will require more. Thus, the ideal solution is assumed to exist in a small subspace of the column space of $\mathbf{\Phi}$. The basis pursuit algorithm discovers this subspace, which motivates its name. By restricting the subspace, it discriminates against vectors

$$\{\mathbf{x} \, : \, ||\mathbf{p} - \mathbf{\Phi}\mathbf{x}||_2 < \sigma\}, \tag{3.3.23}$$

i.e. solutions for **x** that explain the data with the required fidelity, but which are less sparse and hence more likely to be corrupted. The parameter σ controls the data misfit, for cases when it is not desirable to perfectly fit the input data **p**, e.g. when **p** contains noise.

When strictly considering the problem of explaining $\mathbf{P}(z_0,\, z_0)$ in the focal domain, in principle $\sigma = 0$ and there is no clear motivation for minimizing the sum-norm of the focal subdomains, as in principle any solution is equally good, sparse or not. Figure 3.8 depicts slices of focal subdomains that where acquired by solving (3.3.16), as well as a version of (3.3.16) where the sum-norm has been replaced with the Frobenius norm. The sum-norm solution is indeed sparser, yet both solutions explain $\mathbf{P}(z_0,\, z_0)$ equally well, as demonstrated in figure 3.9.

The preference for a sparsity promoting functional becomes more clearly motivated when dealing with a corrupted version of $\mathbf{P}(z_0,\, z_0)$. The BPDN approach was proposed by Kutscha [2014], with the objective of seismic data interpolation. In the interpolation problem, the ideal data is corrupted by undersampling. The sum-norm helps suppress aliasing artifacts in the focal subdomain, yielding an interpolated result where aliasing effects are reduced. Similarly in the deblending problem, unblended data is corrupted by blending and the sum-norm is instrumental in suppressing blending noise in the focal domain. Section 3.5 discusses focal deblending in more depth.

For solving (3.3.16) the spectral projected gradient L1 (SPGL1) solver is used [van den Berg and Friedlander, 2007]. A summary of the algorithm can be found in appendix B. An alternative formulation of focal deblending that uses a greedy solver can be found in Cao et al. [2019]. Using a greedy solver was more efficient when it came to number of iterations, at the cost of a slightly worse deblending result.

3.4 The multilevel single-sided focal transform

The double focal transform, as presented earlier, uses two one-way operators to handle upgoing and downgoing wavefield extrapolation, from the surface to each chosen depth level. Another possibility, which I explore here, is to use one two-way operator that extrapolates from the surface to some depth level z_k and back to the surface. This implies that this new extrapolation operator must include scattering in some form, unlike the case of the one-way operators. The concept of using two-way operators is along the thinking found in Berkhout and Verschuur [2006] and an application on focal deblending is described in Kontakis et al. [2016].

3.4.1 The inverse transform

We begin again from WRW modeling:

$$\mathbf{P}(z_0,\, z_0) = \mathbf{D}(z_0)\sum_{m=1}^{M}[\mathbf{W}_m^-(z_0,\, z_m)\mathbf{R}(z_m,\, z_m)\mathbf{W}_m^+(z_m,\, z_0)]\mathbf{S}^+(z_0).$$

Setting $\mathbf{D}(z_0) = \mathbf{I}$, $\mathbf{R}(z_m,\, z_m) = \mathbf{I}$ and $\mathbf{S}^+(z_0) = \mathbf{X}_k^S(z_0)$ for K depth levels we get

$$\begin{aligned}\mathbf{P}(z_0,\, z_0) &= \sum_{k=1}^{K}[\mathbf{W}_k^-(z_0,\, z_k)\mathbf{W}_k^+(z_k,\, z_0)]\mathbf{X}_k^S(z_0)\\ &= \sum_{k=1}^{K}\mathbf{W}_k^\pm(z_0,\, z_0)\mathbf{X}_k^S(z_0).\end{aligned} \tag{3.4.24}$$

$\mathbf{W}_k^\pm(z_0,\, z_0)$ is perhaps the simplest possible two-way operator. It models propagation through a homogeneous halfspace, defined by a locally-reacting flat reflector at z_k with unit reflectivity everywhere. $\mathbf{X}_k^S(z_0)$ can be understood as a virtual source array that generates a signal that propagates in the halfspace medium and generates a component of $\mathbf{P}(z_0,\, z_0)$. Apart from having a separate two-way operator, each depth level is associated with a different virtual source array. The total $\mathbf{P}(z_0,\, z_0)$ is given by summing all components. I shall refer to this form of the single-sided focal transform as the *source-side* focal transform, since the subdomains $\mathbf{X}_k^S(z_0)$ take the place of the source.

Choosing $\mathbf{D}(z_0) = \mathbf{X}_k^R(z_0)$, $\mathbf{R}(z_m,\, z_m) = \mathbf{I}$ and $\mathbf{S}^+(z_0) = \mathbf{I}$ yields another form of single-sided focal transform,

$$\begin{aligned}\mathbf{P}(z_0,\, z_0) &= \sum_{k=1}^{K}\mathbf{X}_k^R(z_0)[\mathbf{W}_k^-(z_0,\, z_k)\mathbf{W}_k^+(z_k,\, z_0)]\\ &= \sum_{k=1}^{K}\mathbf{X}_k^R(z_0)\mathbf{W}_k^\pm(z_0,\, z_0).\end{aligned} \tag{3.4.25}$$

The two-way operator stays the same, but the virtual source array has been replaced by impulses at the surface. The upgoing wavefield that arrives at the surface from z_k encounters a virtual receiver array $\mathbf{X}_k^R(z_0)$ that produces a wavefield component, which when stacked with the other $K - 1$ components yields $\mathbf{P}(z_0,\, z_0)$. Using the same naming convention, (3.4.25) will be referred to as the *receiver-side* single focal transform.

The three focal subdomains aim to explain the components of $\mathbf{P}(z_0,\, z_0)$ from different points of view (figure 3.10). In the double focal transform the subdomains explain the signal via means of scattering, possibly in combination with residual propagation (for unfocused events). In other words, they encode action that conceptually

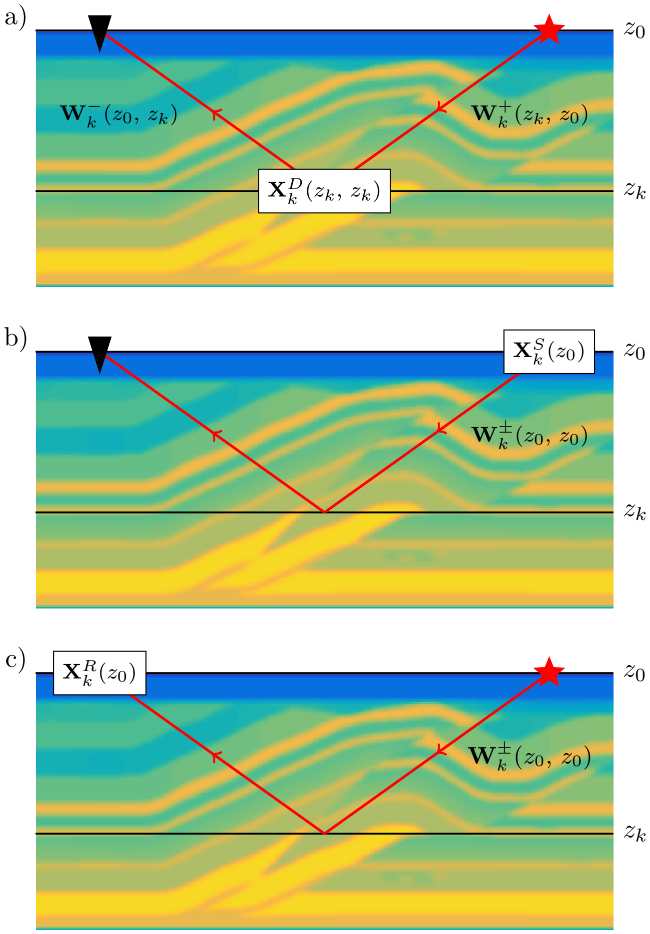


Figure 3.10: *Schematic representation of the different focal transforms, a) double focal transform, b) single focal transform (source-side), c) single-focal transform (receiver side).*

takes place in the subsurface. On the contrary, both variants of the single-sided focal transform encode action that conceptually happens at the surface. Due to this reason, the content of focal subdomains may look quite different, depending on which type was chosen. This can be illustrated by comparing the case of focusing a flat-layer-reflection event (figure 3.11) with that of focusing a wavefield originating from a point diffractor (figure 3.12). For the case of the flat reflector focused by accurate operators, the focal domains look similar (compare figures 3.11d-f with 3.11g-i). This is because the event recorded at the surface can be interpreted equally well as a collection of individual diffractors placed at the same depth level (double focal transform), or as the result of a point source at each shot location generating a wavefield that reflects from a flat reflector, the scattering itself contained within the transform operator (single-sided focal transform). The situation, however, changes when it comes to focusing a single diffraction event. In this case the difference between the two transforms becomes apparent in the focal domain content (compare figures 3.12d-f with 3.12g-i). As expected, the double focal transform focuses to a single point, from which the diffraction is generated, as implied by figures 3.12d-f. The content of the single-sided focal transform, shown in figures 3.12g-i, indicates that an array of virtual sources is needed at the surface. Each source ‘emits’ its estimated signature at a different time, synthesizing the diffraction event. These two examples are stylized, but highlight the fact that the interpretation of the focal domain content is different for the various definitions of the focal transform, despite the fact that in certain cases the focal subdmains may look similar in appearance.

The focusing operation, which amounts to applying the adjoint operator $\mathbf{W}_k^{\pm}(z_0,\, z_0)^{\mathrm{H}}$, i.e.,

$$\boldsymbol{\mathcal{X}}_k^{S}(z_0) = \mathbf{W}_k^{\pm}(z_0,\, z_0)^{\mathrm{H}}\mathbf{P}(z_0,\, z_0) \tag{3.4.26}$$

and

$$\boldsymbol{\mathcal{X}}_k^{R}(z_0) = \mathbf{P}(z_0,\, z_0)\mathbf{W}_k^{\pm}(z_0,\, z_0)^{\mathrm{H}} \tag{3.4.27}$$

reveals another important difference. In (3.4.26) the focusing operation is applied on independent common shot gathers, that is columns of $\mathbf{P}(z_0,\, z_0)$. In (3.4.27) on the other hand, it acts on independent common receiver gathers, i.e. rows of $\mathbf{P}(z_0,\, z_0)$. Contrasting both of these cases, focusing under the double focal transform, as described in the previous section,

$$\boldsymbol{\mathcal{X}}_k^{D}(z_k,\, z_k) = \mathbf{W}_k^{-}(z_0,\, z_k)^{\mathrm{H}}\mathbf{P}(z_0,\, z_0)\mathbf{W}_k^{+}(z_k,\, z_0)^{\mathrm{H}}, \tag{3.4.28}$$

involves both the columns and rows of $\mathbf{P}(z_0,\, z_0)$. The capability of independently treating common shot and common receiver gathers is the basis behind the smart subsets idea discussed in chapter 4.

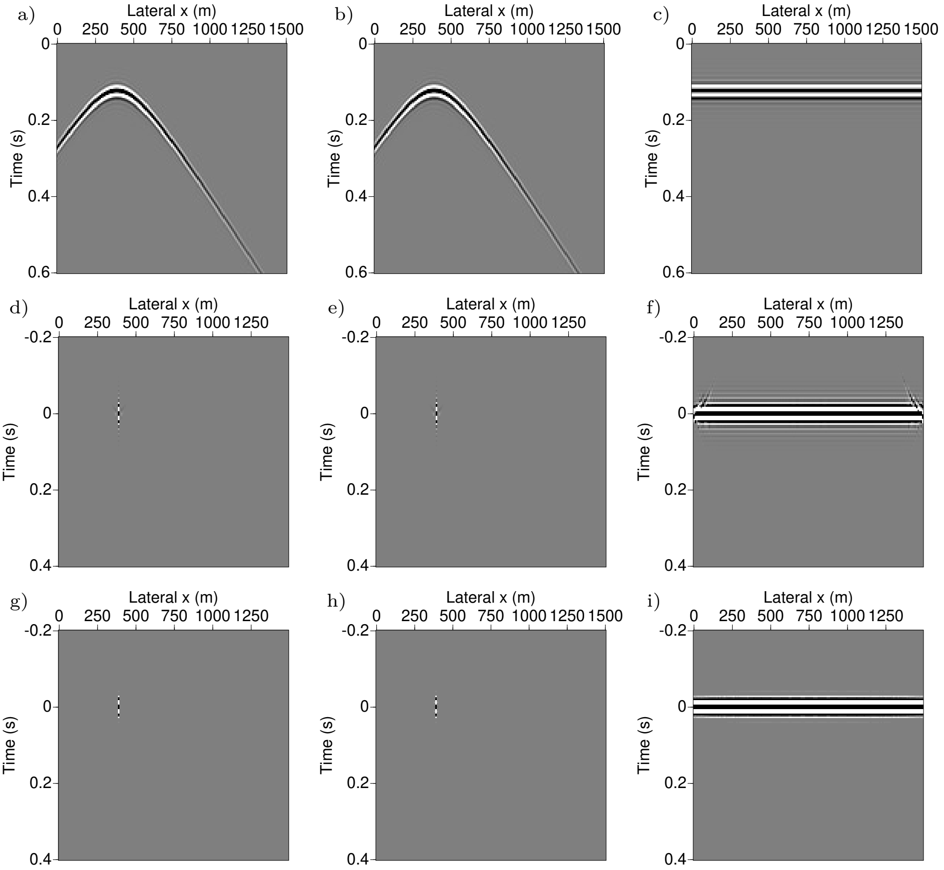


Figure 3.11: *A reflection event from a flat layer and its representation in different focal transforms. Top row: event to be focused; a) common shot gather, b) common receiver gather, c) zero offset gather. Middle row: focal subdomain, double focal transform; d) common focal grid point gather, e) common local offset gather, f) subsurface zero offset gather. Bottom row: focal subdomain, single-sided focal transform (source-side); g) common virtual shot gather, h) common receiver gather, i) zero offset gather.*

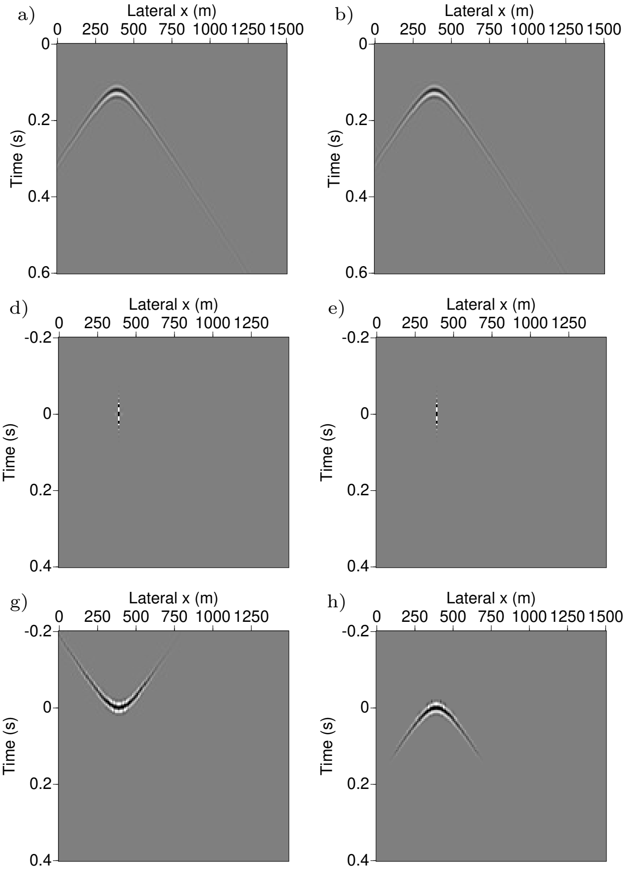


Figure 3.12: *A diffraction event and its representation in different focal transforms. Top row: diffraction to be focused; a) common shot gather, b) common receiver gather, c) zero offset gather. Middle row: focal subdomain, double focal transform; d) common focal grid point gather, e) common local offset gather, f) subsurface zero offset gather. Bottom row: focal subdomain, single-sided focal transform (source-side); g) common virtual shot gather, h) common receiver gather, i) zero offset gather.*

■ 3.4.2 The forward transform

The forward transform is again given by solving an optimization problem. For the source-side transform, the focal subdomains are given by

$$\begin{aligned}&\left\{\hat{\mathbf{X}}^S_{\mathrm{rec},1}(z_1),\, \hat{\mathbf{X}}^S_{\mathrm{rec},2}(z_2),\, \ldots,\, \hat{\mathbf{X}}^S_{\mathrm{rec},K}(z_K)\right\} = \\ &\underset{\substack{\hat{\mathbf{X}}^S_k(z_0)\\ k=1,\ldots,K}}{\arg\min} \left\{\sum_t \sum_{k=1}^{K} \left|\left|\hat{\mathbf{X}}^S_k(z_0)\right|\right|_{\mathrm{S}}\right\} \quad \text{subject to} \\ &\sum_\omega \left|\left|\mathbf{P}(z_0,\, z_0) - \sum_{k=1}^{K} \mathbf{W}^{\pm}_k(z_0,\, z_0)\mathbf{X}^S_k(z_0)\right|\right|_{\mathrm{F}} \leq \sigma.\end{aligned} \tag{3.4.29}$$

For the receiver-side transform it predictably becomes,

$$\begin{aligned}&\left\{\hat{\mathbf{X}}^R_{\mathrm{rec},1}(z_1),\, \hat{\mathbf{X}}^R_{\mathrm{rec},2}(z_2),\, \ldots,\, \hat{\mathbf{X}}^R_{\mathrm{rec},K}(z_K)\right\} = \\ &\underset{\substack{\hat{\mathbf{X}}^R_k(z_0)\\ k=1,\ldots,K}}{\arg\min} \left\{\sum_t \sum_{k=1}^{K} \left|\left|\hat{\mathbf{X}}^R_k(z_0)\right|\right|_{\mathrm{S}}\right\} \quad \text{subject to} \\ &\sum_\omega \left|\left|\mathbf{P}(z_0,\, z_0) - \sum_{k=1}^{K} \mathbf{X}^R_k(z_0)\mathbf{W}^{\pm}_k(z_0,\, z_0)\right|\right|_{\mathrm{F}} \leq \sigma.\end{aligned} \tag{3.4.30}$$

This concludes the discussion on the various forms of the focal transform, leading to the next topic of applying the tools discussed so far for deblending purposes.

3.5 Formulation of focal deblending

Focal deblending [Kontakis and Verschuur, 2014] combines the blending equation, introduced in chapter 2, with the focal transforms introduced earlier in this chapter. The blending equation (2.2.11), rewritten for $z_r = z_s = z_0$, becomes

$$\mathbf{P}_{\mathrm{bl}}(z_0,\, z_0) = \mathbf{P}(z_0,\, z_0)\mathbf{\Gamma}(z_0). \tag{3.5.31}$$

We assume that there exists a set of K focal subdomains $\mathbf{X}^D_k(z_k,\, z_k)$, such that, using the double focal transform,

$$\mathbf{P}(z_0,\, z_0) = \sum_{k=1}^{K} \mathbf{W}^{-}_k(z_0,\, z_k)\mathbf{X}^D_k(z_k,\, z_k)\mathbf{W}^{+}_k(z_k,\, z_0). \tag{3.5.32}$$

Plugging this into (3.5.31) yields

$$\mathbf{P}_{\mathrm{bl}}(z_0,\, z_0) = \sum_{k=1}^{K} \left[\mathbf{W}^{-}_k(z_0,\, z_k)\mathbf{X}^D_k(z_k,\, z_k)\mathbf{W}^{+}_k(z_k,\, z_0)\right]\mathbf{\Gamma}(z_0). \tag{3.5.33}$$

Deblending then becomes a problem of solving for the focal subdomains $\mathbf{X}_k^D(z_k,\, z_k)$, rather than for $\mathbf{P}(z_0,\, z_0)$ directly. The purpose of this detour is to leverage focusing and the sparse representation of seismic events in the focal domain to discriminate against solutions that contain blending noise. The assumption being made here is that solutions $\hat{\mathbf{X}}_{\mathrm{debl},k}^D(z_k,\, z_k)$ that are free of blending noise are sparser than those that include it. We promote such sparse solutions by solving the BPDN problem

$$\begin{gathered}\left\{\hat{\mathbf{X}}_{\mathrm{debl},1}^D(z_1,\, z_1),\, \hat{\mathbf{X}}_{\mathrm{debl},2}^D(z_2,\, z_2),\, \ldots,\, \hat{\mathbf{X}}_{\mathrm{debl},K}^D(z_K,\, z_K)\right\} = \\ \underset{\substack{\hat{\mathbf{X}}_k^D(z_k,\, z_k) \\ k=1,\ldots,K}}{\arg\min} \left\{\sum_t \sum_{k=1}^K \left|\left|\hat{\mathbf{X}}_k^D(z_k,\, z_k)\right|\right|_{\mathrm{S}}\right\} \quad \text{subject to} \\ \sum_\omega \left|\left|\mathbf{P}(z_0,\, z_0) - \sum_{k=1}^K \mathbf{W}_k^-(z_0,\, z_k)\mathbf{X}_k^D(z_k,\, z_k)\mathbf{W}_k^+(z_k,\, z_0)\mathbf{\Gamma}(z_0)\right|\right|_{\mathrm{F}} \leq \sigma. \end{gathered} \tag{3.5.34}$$

Extracting the deblended wavefield amounts to inverse transforming the focal subdomains, i.e.

$$\mathbf{P}_{\mathrm{debl}}(z_0,\, z_0) = \sum_{k=1}^K \mathbf{W}_k^-(z_0,\, z_k)\mathbf{X}_{\mathrm{debl},k}^D(z_k,\, z_k)\mathbf{W}_k^+(z_k,\, z_0). \tag{3.5.35}$$

Unlike the case of the forward focal transform as defined in (3.3.16), the motivation for using a sparsifying functional such as the sum-norm is clear here, as it is what enables blending noise suppression. This can be easily demonstrated with an example, shown in figure 3.13. When the Frobenius norm is used in the objective function (figure 3.13b), blending noise is not suppressed effectively in the focal domain, which means that it will be inevitably present in the deblended result. On the other hand, the sum-norm solution (figure 3.13c) effectively suppresses most of the blending noise in the focal domain, leading to a cleaner result.

The vectorized form of (3.5.34) is given by

$$\mathbf{x}_{\mathrm{debl}} = \underset{\mathbf{x}}{\arg\min} \left\{||\mathbf{x}||_1\right\} \quad \text{subject to} \quad ||\mathbf{p}_{\mathrm{bl}} - \mathbf{\Psi\Phi x}||_2 \leq \sigma, \tag{3.5.36}$$

where $\mathbf{p}_{\mathrm{bl}}$ and $\mathbf{x}_{\mathrm{debl}}$ are the vectorized blended data and deblended focal subdomain data cubes. $\mathbf{\Psi}$ is a matrix that carries out the blending operation. This form of the problem highlights the *compressive sensing* nature of the problem, where $\mathbf{\Psi}$ acts as a measurement matrix, the inner product of each row with $\mathbf{y} = \mathbf{\Phi x}$ generating a blended sample. As before, $\mathbf{\Phi}$ plays the role of a sparsifying dictionary. The relationship between (3.5.34) and (3.5.36) is explored in more detail in appendix A.

Deblending results can be often improved by taking advantage of the structure found in the focal subdomains. Focused events tend to have the majority of their amplitude concentrated near the origin in a time and offset coordinate system. Content away

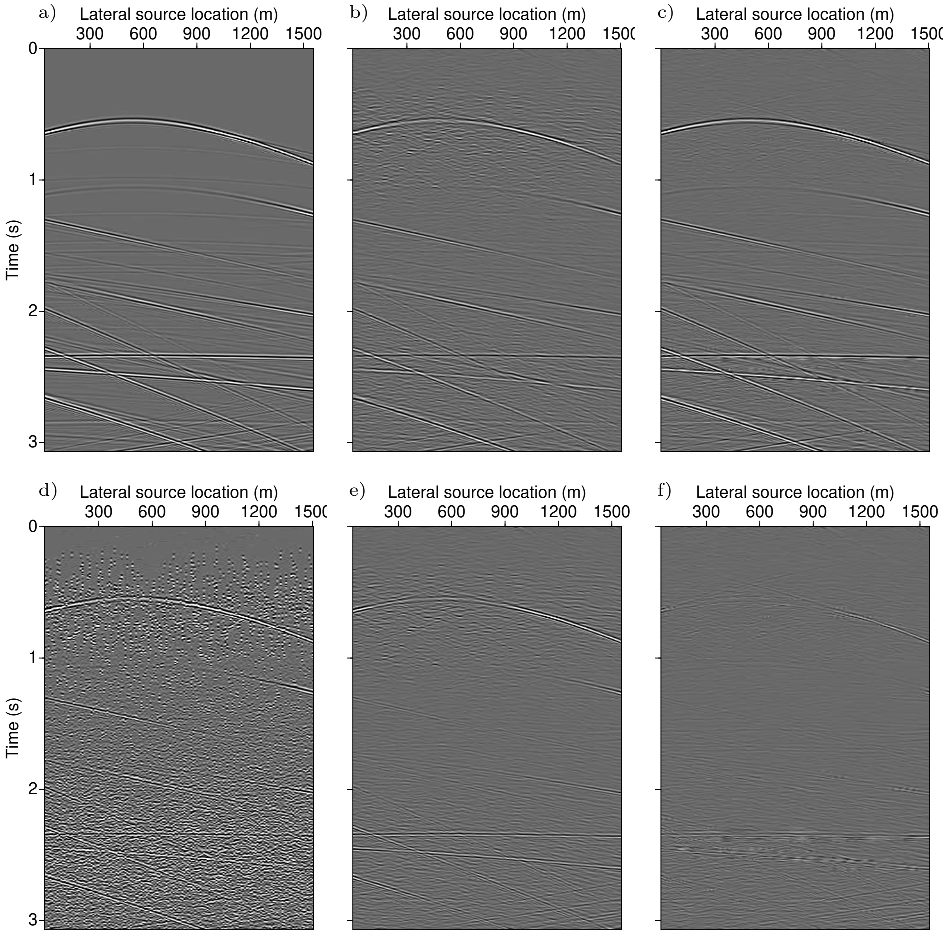


Figure 3.13: *Comparison of deblending results when using different norms in the objective function of problem (3.5.34); a) unblended data, b) deblended result when using the Frobenius norm, c) deblended result when using the sum-norm, d) pseudo-deblended data, e) difference between unblended data and the Frobenius result, f) difference between unblended data and the sum-norm result. The sum-norm suppresses most of the blending noise. A modified version of the SEAM overthrust model discussed in section 4.1 was used to generate the synthetic data for this example.*

from the origin is more likely to represent blending noise, or uncompressed events to be handled in other focal subdomains. By penalizing amplitudes away from the origin, more blending noise can be suppressed in the solution.

A straightforward way to include such a penalizing scheme is to replace the sum-norm in (3.5.34) with a weighted sum-norm, i.e. $||\hat{\mathbf{X}}_k^D(z_k,\, z_k)||_S$ is replaced by $||\hat{\mathbf{L}}_k \odot \hat{\mathbf{X}}_k^D(z_k,\, z_k)||_S$. $\hat{\mathbf{L}}_k$ is the cube array of weights to be applied to the k-th focal subdomain and $\odot$ is the Hadamard product (element-by-element multiplication). For the vectorized form of the problem, $||\mathbf{x}||_1$ in problem (3.5.36) is replaced by $||\mathbf{\Lambda}\mathbf{x}||_1$, where $\mathbf{\Lambda}$ is a diagonal matrix that holds the weights.

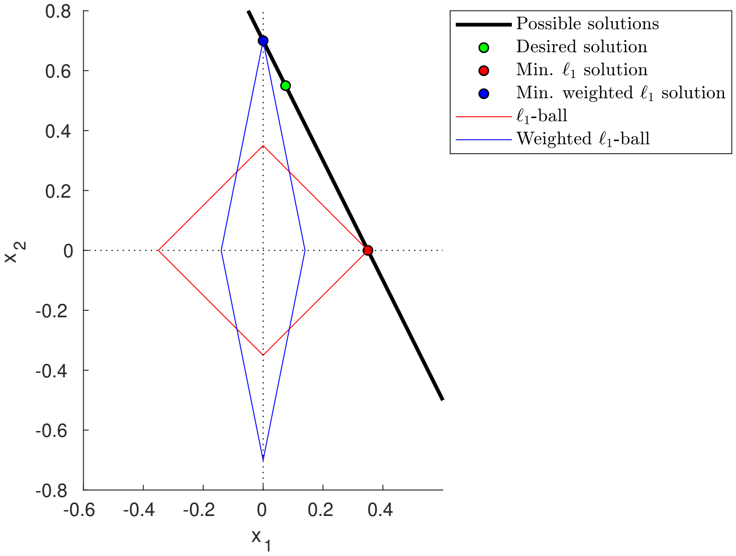


Figure 3.14: *Using a weighted ℓ_1-norm in a BPDN problem may lead to a more desired solution.*

The goal of BPDN is to find a solution with minimal ℓ_1-norm. This is equivalent to finding the ℓ_1-ball with minimal radius which touches the translated nullspace of $\mathbf{\Psi\Phi}$. An example with two variables is given in figure 3.14. The weights dilate/contract the shape of the ℓ_1-ball along the various dimensions, which leads to different solutions that may be more preferrable. These weights provide an additional way to bring in prior knowledge about the desired solution. An example is illustrated in figure 3.15. The same dataset as the one used in figure 3.13 was blended with a blending factor equal to 4. Two deblended results were calculated, the first using the unweighted sum-norm and the second using a weighted sum norm with radially increasing penalty centered at zero time/zero local offset. The weights can

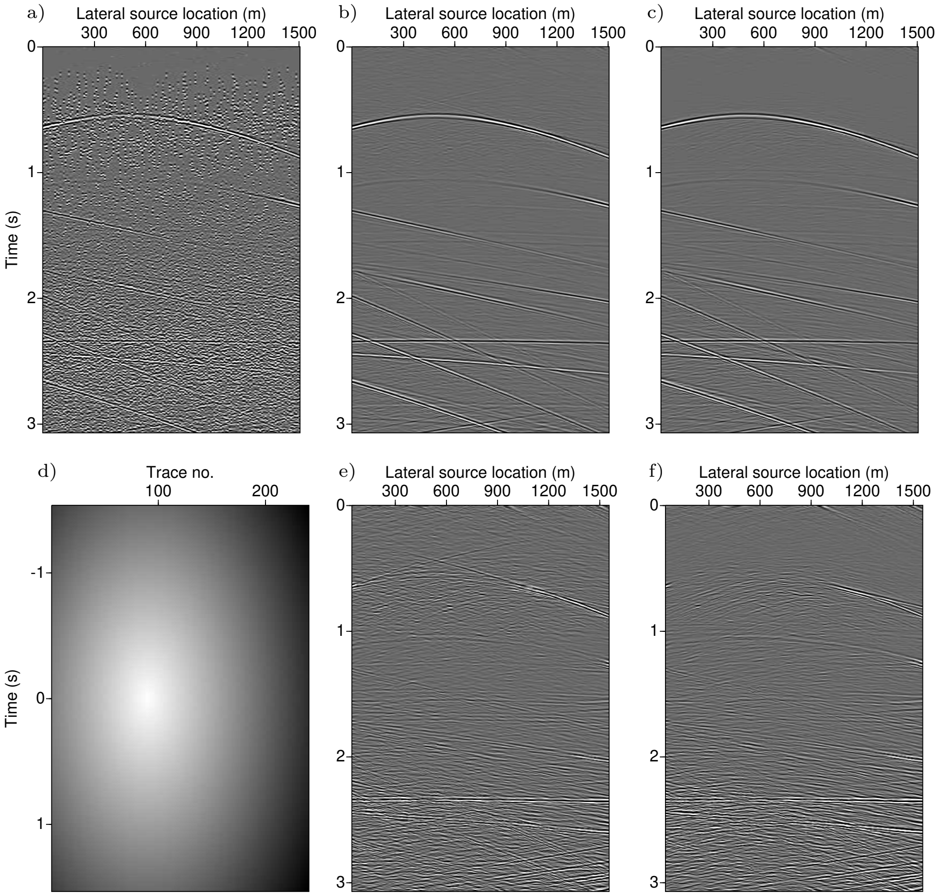


Figure 3.15: *a) Pseudo-deblended data, b) deblended using the unweighted sum-norm, c) deblended using the weighted sum-norm, d) the weights used for the sum-norm (darker color denotes higher penalty), e) difference of b) from the ideal result (clipped), f) difference of c) from the ideal result (clipped).*

be seen in figure 3.15d. Comparing the difference of the deblended results from the ideal unblended data shows less blending noise leakage when the weighted sum-norm is used. The downside of using weighting is that it can have the opposite effect of hampering the result when the assumptions about the structure of signal and noise in the focal domain are not met.

3.6 Sparsity and blending noise leakage

The deblending process is usually not perfect, leading to separated wavefields that suffer from various degrees of blending noise leakage. This raises two important questions, namely what are the causes of imperfect separation and under which conditions is the deblending result perfect. In this section I discuss two main sources of separation errors. The first is related to the limited range of the focal operators. The second is related to the sparsity assumption used in the deblending process.

■ 3.6.1 Blending noise leakage due to the limited range of focal operators

When using a dictionary $\mathbf{\Phi}$, writing

$$\mathbf{p} = \mathbf{\Phi}\mathbf{x} \tag{3.6.37}$$

implies that the unblended data $\mathbf{p}$ can be expressed as a linear combination of the atoms of the dictionary $\mathbf{\Phi}$. Or, in other words, we assume that $\mathbf{p}$ is in the range of $\mathbf{\Phi}$. If $\mathbf{\Phi}$ forms a frame or basis for all possible $\mathbf{p}$, then (3.6.37) will always hold. It is possible, however, that $\mathbf{\Phi}$ does not span the entire space of possible $\mathbf{p}$ vectors. In that case, there does not exist an $\mathbf{x}$ such that (3.6.37) holds and one can only hope for an acceptable approximation. It will be shown below that this can create the additional problem of blending noise leakage. This will be demonstrated in two steps. The first step is to show that when $\mathbf{\Phi}$ implements the focal transform, there exist $\mathbf{p}$ that cannot be fully reconstructed. The second step is to show that the component of $\mathbf{p}$ that cannot be reconstructed may become the source of blending noise contamination in a deblending problem.

For the first step, I begin with a continuous form of the inverse double focal transform. As was the case previously, I assume M focal operators that represent one-way extrapolation in a homogeneous, acoustic, isotropic medium between the surface and a flat layer in the subsurface. Then, the value of the field at $\mathbf{r}$ by a source at $\mathbf{s}$ at frequency ω, is given by

$$\mathbf{p}(\mathbf{r}, \mathbf{s}; \omega) = \sum_{m=1}^{M} \iint\limits_{-\infty}^{+\infty} w_m^-(\mathbf{r}, \mathbf{u}'; \omega) x_m(\mathbf{u}', \mathbf{u}, \omega) w_m^+(\mathbf{u}, \mathbf{s}; \omega) \mathrm{d}\mathbf{u}' \mathrm{d}\mathbf{u}, \tag{3.6.38}$$

where $w_m^{\mp}(\mathbf{r},\mathbf{u}';\omega)$ are the focal operators and $x_m(\mathbf{u}',\mathbf{u},\omega)$ is the focal subdomain. Since the value of the focal operators depends only on the differences $\mathbf{r}-\mathbf{u}'$ and $\mathbf{u}-\mathbf{s}$, (3.6.38) is in fact a linear convolution,

$$\mathbf{p}(\mathbf{r},\mathbf{s};\omega)=\sum_{m=1}^{M}\iint\limits_{-\infty}^{+\infty} w_m^{-}(\mathbf{r}-\mathbf{u}';\omega)x_m(\mathbf{u}',\mathbf{u},\omega)w_m^{+}(\mathbf{u}-\mathbf{s};\omega)\mathrm{d}\mathbf{u}'\mathrm{d}\mathbf{u}. \qquad (3.6.39)$$

Applying the Fourier transform with respect to the spatial coordinates and using the convolution theorem for the Fourier transform gives equation (3.6.39) in the FK domain,

$$\breve{p}(\mathbf{k}_r,\mathbf{k}_s;\omega)=\sum_{m=1}^{M}\breve{w}_m^{-}(\mathbf{k}_r;\omega)\breve{x}_m(\mathbf{k}_r,\mathbf{k}_s;\omega)\breve{w}_m^{+}(\mathbf{k}_s;\omega), \qquad (3.6.40)$$

where $\mathbf{k}_r$ and $\mathbf{k}_s$ are wavevectors with $||\mathbf{k}_r||_2=||\mathbf{k}_s||_2=\omega/c_m$, c_m being the acoustic velocity of the m-th propagator. The quantities $\breve{w}_m^{\mp}(\mathbf{k}_{\{r,s\}};\omega)$ and $\breve{x}_m(\mathbf{k}_r,\mathbf{k}_s;\omega)$ are the Fourier-transformed versions of $w_m^{\mp}(\{\mathbf{r},\mathbf{s}\};\omega)$ and $x_m(\mathbf{u}',\mathbf{u},\omega)$ respectively:

$$\breve{w}_m^{-}(\mathbf{k}_r;\omega)=\int\limits_{-\infty}^{+\infty} w_m^{-}(\mathbf{r};\omega)e^{j\mathbf{k}_r\cdot\mathbf{r}}\mathrm{d}\mathbf{r}, \qquad (3.6.41)$$

$$\breve{w}_m^{+}(\mathbf{k}_s;\omega)=\int\limits_{-\infty}^{+\infty} w_m^{+}(\mathbf{r};\omega)e^{j\mathbf{k}_s\cdot\mathbf{s}}\mathrm{d}\mathbf{s}, \qquad (3.6.42)$$

$$\breve{x}_m(\mathbf{k}_r,\mathbf{k}_s;\omega)=\iint\limits_{-\infty}^{+\infty} x_m(\mathbf{u}',\mathbf{u},\omega)e^{j(\mathbf{k}_r\cdot\mathbf{r}+\mathbf{k}_s\cdot\mathbf{s})}\mathrm{d}\mathbf{u}'\mathrm{d}\mathbf{u}. \qquad (3.6.43)$$

Let

$$k_{\{r,s\},z}=\sqrt{\omega^2/c_m^2-k_{\{r,s\},x}^2-k_{\{r,s\},y}^2} \qquad (3.6.44)$$

be the vertical z-component of $\mathbf{k}_{\{r,s\}}$ and

$$\mathbf{k}_{\{r,s\}}=[k_{\{r,s\},x},\, k_{\{r,s\},y},\, k_{\{r,s\},z}]^{\mathrm{T}}, \qquad (3.6.45)$$

for 3D propagation. In the 2D case either the x- or y-component vanishes. For a homogeneous acoustic medium,

$$\breve{w}_m^{\mp}(\mathbf{k}_r;\omega)=e^{-jk_{\{r,s\},z}|z_m-z_0|}. \qquad (3.6.46)$$

Using (3.6.44), (3.6.46) and (3.6.40),

$$\breve{p}(\mathbf{k}_r,\mathbf{k}_s;\omega)=\sum_{m=1}^{M}e^{-jk_{r,z}|z_m-z_0|}e^{-jk_{s,z}z}\breve{x}_k(\mathbf{k}_r,\mathbf{k}_s;\omega)$$

$$= \sum_{m=1}^{M} e^{-j\sqrt{(\omega/c_m)^2-(k_{r,x}^2+k_{r,y}^2)}|z_m-z_0|}$$
$$\times\, e^{-j\sqrt{(\omega/c_m)^2-(k_{s,x}^2+k_{s,y}^2)}|z_m-z_0|}\breve{x}_m(\mathbf{k}_r,\mathbf{k}_s;\omega). \tag{3.6.47}$$

Note that when either $k_{r,x}^2 + k_{r,y}^2 > \omega^2/c_m^2$ or $k_{s,x}^2 + k_{s,y}^2 > \omega^2/c_m^2$, the exponent becomes a negative real number. For wavenumbers above a cutoff given by ω/c_m, the focal operator becomes evanescent. Plane waves with associated wavenumbers above the cutoff are not extrapolated by the operator. They are exponentially attenuated instead, the rate depending on z_m and ω. Figure 3.16 shows an example of a 2D focal operator for $z = 30$ m and $c = 1500$ m s^{-1}. As can be clearly seen, the magnitude of the operator tends quickly to 0 after the cutoff wavenumber. Although the magnitude technically never becomes zero except at infinity, in practice the value quickly becomes too small to be represented with finite precision arithmetic. Essentially the focal operator acts as a lowpass filter in the FK domain, the evanescent part acting as the stopband of the filter. This gives the focal operators a nullspace: any seismic event composed of plane waves with wavenumbers k_w that satisfy

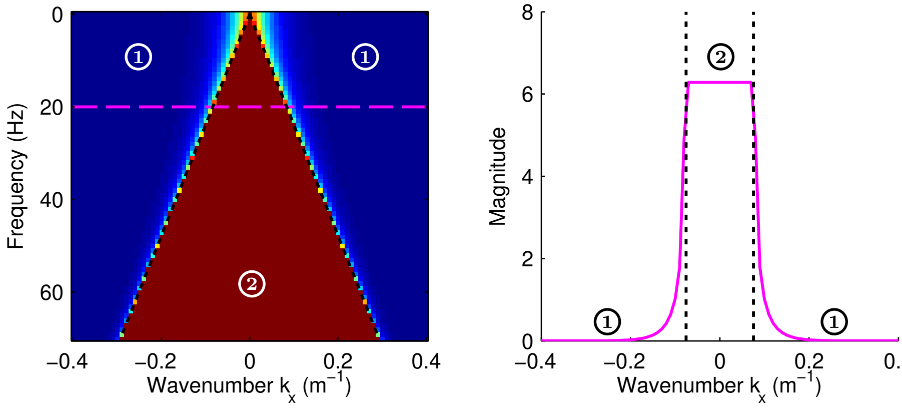


Figure 3.16: *Left: Magnitude of the FK spectrum of a focal operator for different values of frequency and wavenumber k_x. Right: The magnitude of the operator at* 20 *Hz. The marks 1 and 2 denote the evanescent and propagating parts of the spectrum, respectively.*

$$k_w > \omega/\min\{\{c_1, c_2, \ldots, c_M\}\}, \tag{3.6.48}$$

is not in the span of the focal operators. In other words, the span of the focal transform as a whole is limited by the focal operator with the lowest velocity.

If care is not taken, this bandwidth limitation can affect the reconstruction of certain events, such as slow direct waves. Figure 3.17 shows such an example. Figures

3.17a and 3.17b show the same synthetic data in the TX and the FK domain. Reconstructed data using erroneous velocities for the focal operators are shown in figures 3.17c and 3.17d. Figures 3.17e and 3.17f show the differences. Because the velocity used for the ‘slowest’ focal operator was higher than that of the direct wave, the direct wave was not reconstructed.

Apart from inability to reconstruct certain seismic events, the limited range of the focal operators can pose an additional problem when used in a deblending setting, which leads us to the second part of the discussion. Here the assumption is made that $\mathbf{\Gamma}(z_0)$ implements a random time delay code, that every shot participates in one blended shot gather and that each blended shot gather is the result of blending at least two shot gathers. I also define the blending-noise operator,

$$\mathbf{N}(z_0) \equiv (\mathbf{\Gamma}(z_0)\mathbf{\Gamma}(z_0)^{\mathrm{H}} - \mathbf{I})\mathbf{\Lambda}(z_0), \tag{3.6.49}$$

a modified version of the pseudo-deblending operator, $\mathbf{\Gamma}(z_0)\mathbf{\Gamma}(z_0)^{\mathrm{H}}$. Due to the fact that the diagonal elements of $\mathbf{N}(z_0)$ are equal to zero for every frequency slice, the product $\mathbf{P}(z_0,\, z_0)\mathbf{N}(z_0)$ produces only the blending noise component. This is unlike pseudo-deblending, which produces a mixture of both data and blending noise. Each frequency slice of $\mathbf{\Lambda}(z_0)$ is the same diagonal matrix. Its purpose is to apply a normalization on the columns of $\mathbf{\Gamma}(z_0)\mathbf{\Gamma}(z_0)^{\mathrm{H}} - \mathbf{I}$ that accounts for the potentially different number of unblended shot gathers summed to produce a given blended shot gather. The proper definition of $\mathbf{\Lambda}(z_0)$ can be found in appendix C. An important property of $\mathbf{N}(z_0)$ as defined above is that

$$\mathbf{N}(z_0)\mathbf{\Gamma}(z_0) = \mathbf{\Gamma}(z_0). \tag{3.6.50}$$

The proof of this property is also contained in appendix C.

For a given unblended data $\mathbf{P}(z_0,\, z_0)$ and a definition of the focal transform, we can split $\mathbf{P}(z_0,\, z_0)$ as the sum of two components,

$$\mathbf{P}(z_0,\, z_0) = \mathbf{P}_{\mathcal{R}}(z_0,\, z_0) + \mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0). \tag{3.6.51}$$

$\mathbf{P}_{\mathcal{R}}(z_0,\, z_0)$ is the component which is in the range of the focal transform and $\mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)$ is the component which isn’t. Although by assumption $\mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)$ itself cannot be reconstructed by the focal transform, it could be that the blending noise associated with it, $\mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)\mathbf{N}(z_0)$ is in the range of the focal transform. When that is the case, there exist $\mathbf{X}^{D}_{\mathcal{R},0}(z_0,\, z_0)$ and $\mathbf{X}^{D}_{N,m}(z_m,\, z_m)$ such that

$$\begin{aligned}\mathbf{P}_{\mathcal{R}}(z_0,\, z_0) &= \sum_{m=1}^{M} \mathbf{W}^{-}_{m}(z_0,\, z_m)\mathbf{X}^{D}_{\mathcal{R},m}(z_m,\, z_m)\mathbf{W}^{+}_{m}(z_m,\, z_0),\\ \mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)\mathbf{N}(z_0) &= \sum_{m=1}^{M} \mathbf{W}^{-}_{m}(z_0,\, z_m)\mathbf{X}^{D}_{N,m}(z_m,\, z_m)\mathbf{W}^{+}_{m}(z_m,\, z_0).\end{aligned} \tag{3.6.52}$$

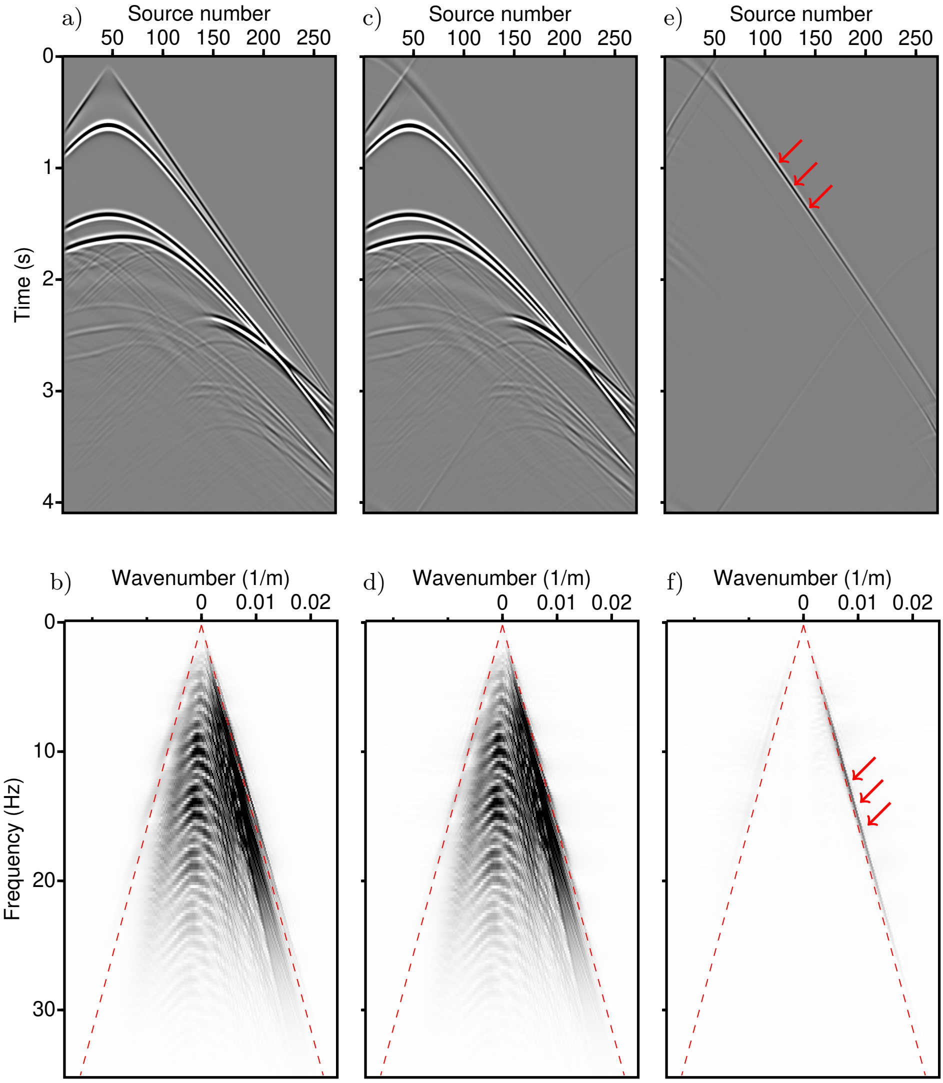


Figure 3.17: *Reconstruction using the focal transform: the original synthetic data in the a) TX domain, b) FK domain; the reconstructed data in the c) TX domain, d) FK domain; the difference of the original from the reconstructed data in the e) TX domain, f) FK domain. Indicated with red arrows is the part of the direct wave that has not been reconstructed.*

I also define

$$\mathbf{P}'(z_0,\, z_0) = \mathbf{P}_{\mathcal{R}}(z_0,\, z_0) + \mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)\mathbf{N}(z_0). \tag{3.6.53}$$

Note that because $\mathbf{N}(z_0) \neq \mathbf{I}$, $\mathbf{P}'(z_0,\, z_0) \neq \mathbf{P}(z_0,\, z_0)$. It is now easy to show that $\mathbf{P}'(z_0,\, z_0)$ also satisfies the blending equation,

$$\begin{aligned}
\mathbf{P}'(z_0,\, z_0)\mathbf{\Gamma}(z_0) &= [\mathbf{P}_{\mathcal{R}}(z_0,\, z_0) + \mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)\mathbf{N}(z_0)]\mathbf{\Gamma}(z_0) \\
&= \mathbf{P}_{\mathcal{R}}(z_0,\, z_0)\mathbf{\Gamma}(z_0) + \mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)\mathbf{N}(z_0)\mathbf{\Gamma}(z_0) \\
&= \mathbf{P}_{\mathcal{R}}(z_0,\, z_0)\mathbf{\Gamma}(z_0) + \mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)\mathbf{\Gamma}(z_0) \\
&= [\mathbf{P}_{\mathcal{R}}(z_0,\, z_0) + \mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)]\mathbf{\Gamma}(z_0) \\
&= \mathbf{P}(z_0,\, z_0)\mathbf{\Gamma}(z_0) \\
&= \mathbf{P}_{\mathrm{bl}}(z_0,\, z_0).
\end{aligned} \tag{3.6.54}$$

The above discussion can be summarized as follows. When using the focal transform, the deblended answer can never be the desired $\mathbf{P}(z_0,\, z_0)$, assuming $\mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0) \neq \mathbf{0}$. This is because in that case $\mathbf{P}_{\mathrm{bl}}(z_0,\, z_0) \neq \mathbf{P}_{\mathcal{R}}(z_0,\, z_0)\mathbf{\Gamma}(z_0)$. On the contrary, the noise-containing $\mathbf{P}'(z_0,\, z_0)$ does satisfy the blending equation, and as such is a candidate solution as long as a) $\mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)\mathbf{N}(z_0)$ is in the range of the focal transform and b) additional constraints do not prohibit it. From these we can conclude that $\mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)$ can only be explained as blending noise. Due to the wideband nature of blending noise in the FK domain, usually only part of $\mathbf{P}_{\bar{\mathcal{R}}}(z_0,\, z_0)\mathbf{N}(z_0)$ will be within the range of the focal operators. In that case, only the part that is within the range will be explained and the rest will stay in the residual, i.e. the difference between the observed blended data and the reblended inversion result.

This type of blending noise leakage is not particular to focal deblending. In principle it can occur in any method that cannot fully explain the unblended data. That being said, the filtering action of the focal operators also has a positive effect: any blending noise which is outside of the ‘propagation cone’ of the focal operator cannot be part of the solution, therefore, it can only be explained as signal or stay in the residual.

Figure 3.18 shows an example of blending noise leakage. The input data and focal operators used are the same as those of the reconstruction experiment discussed earlier. The blended data can be seen in the TX and FK domains in figures 3.18a and 3.18b. The deblending result shown in figures 3.18c and 3.18d shows a positive consequence of the filtering effect of the focal transform. Blending noise outside that falls in the evanescent region of the operator does not become part of the deblended result. On the other hand though, the deblended result is not completely noise-free, as can be witnessed in figures 3.18e and 3.18f. This noise is within the propagating part of the focal operator and corresponds to part of the blending noise representation of the direct wave. As the direct wave cannot be reconstructed, the only way to remove its contribution to the residual is to explain it as blending noise, which leads to the bandpass-filtered blending noise seen in figure 3.18f.

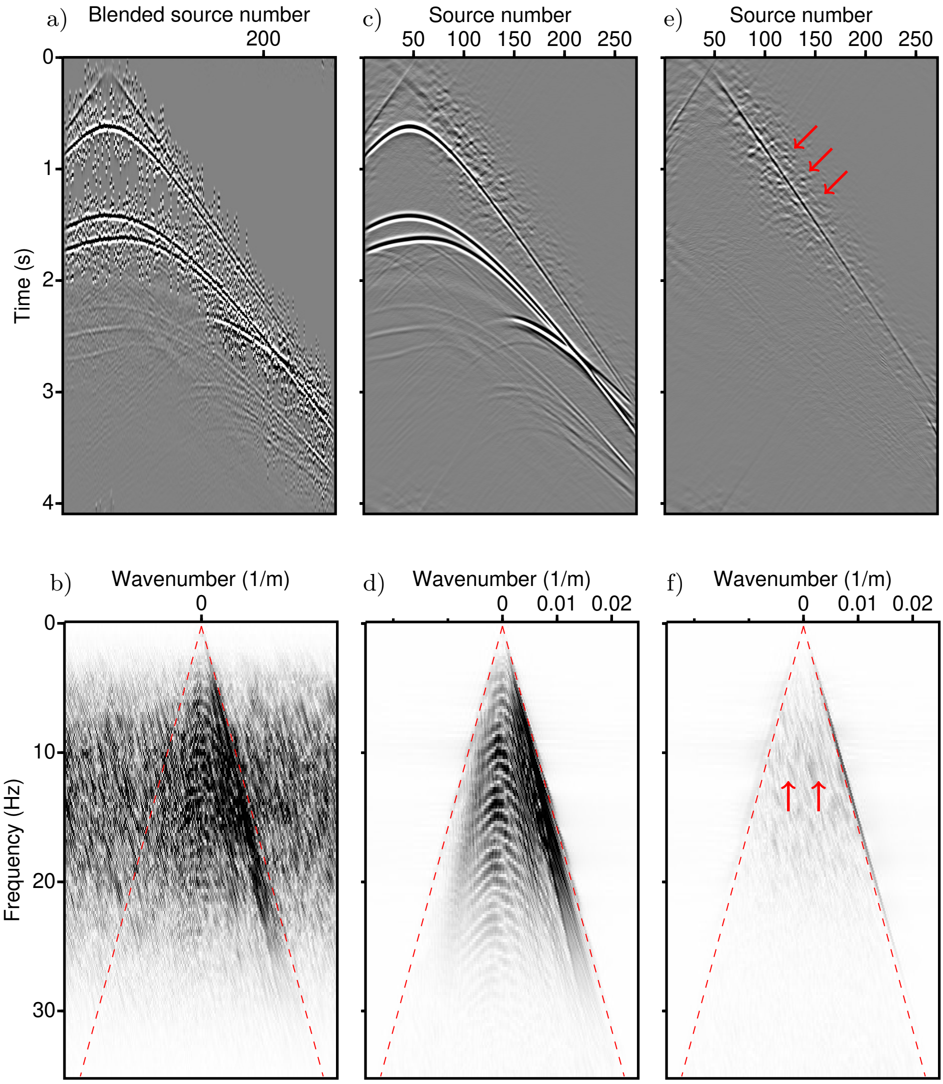


Figure 3.18: *Deblending using the focal transform: blended synthetic data in the a) TX domain, b) FK domain; the deblended data in the c) TX domain, d) FK domain; the difference of the deblended and unblended data in the e) TX domain, f) FK domain. The red arrows point at blending noise leakage.*

If the inability to explain the direct wave is the sole reason blending noise leaks into the result shown in figures 3.18c,d, we would expect that absence of the direct wave would lead to absence of blending noise in the result. To verify this, the deblending experiment is repeated with a small modification: the data is blended after filtering out the direct wave. The same blending code is used as well as the same deblending parameters. The blended data and their FK spectrum are shown in figure 3.19a and 3.19b. The deblended result in 3.19c,d shows hardly any visible blending noise, a fact that is verified by the difference plots in figures 3.19e,f. In this particular case, it was possible to achieve good separation as long as all data could be explained by our choice of transform domain. This is not always possible, however, as good separation will depend on how sparse is the focal domain representation of the result. This source of blending noise leakage is explored in the following section.

■ 3.6.2 Blending noise leakage and the sparsity assumption

By casting deblending as a BPDN problem, I have made the assumption that out of all possible solutions to the blending equation, the one that is free of blending noise will have the minimum ℓ_1-norm. The ℓ_1-norm was used as a convex approximation to the ℓ_0-norm to make the optimization problem computationally tractable. In this sense, the value of the ℓ_1-norm of the solution is used as a proxy for measuring the sparsity of the solution. One may wonder whether the ideal result having minimum ℓ_1-norm is a property that is affected by how sparsely this ideal deblended result is expressed in terms of atoms of $\mathbf{\Phi}$.

To get a feeling for the answer, the following numerical experiment is performed. First, the forward focal transform of a dataset is calculated. The focal subdomain is then thresholded such that all elements of the focal domain are set to zero except the N elements with the highest magnitude. The thresholded focal domain is then inverse transformed, yielding our new input data, and the result is blended and deblended. The quality of the result is measured using the metric (2.4.25). The process is repeated for successively larger values of N.

Figure 3.20a shows the deblending quality versus the ratio of nonzero focal domain elements to the number of blended samples. The quality is over 30dB for a ratio of up to approximately 10% and gradually drops as the ratio becomes bigger. Clearly the deblended result is closer to ideal unblended result the more sparsely the ideal result can be represented in the focal domain. Figure 3.20b shows a comparison between the ℓ_1-norm of the ideal solution and the ℓ_1-norm of the deblended solution. As the ratio of nonzero focal domain elements to number to blended samples increases, the ℓ_1-norm of the deblended solution starts becoming noticeably lower than that of the ideal solution. Progressively, the assumption of minimum ℓ_1-norm for the ideal solution becomes less realistic and the recovered solution moves further away from the ideal one.

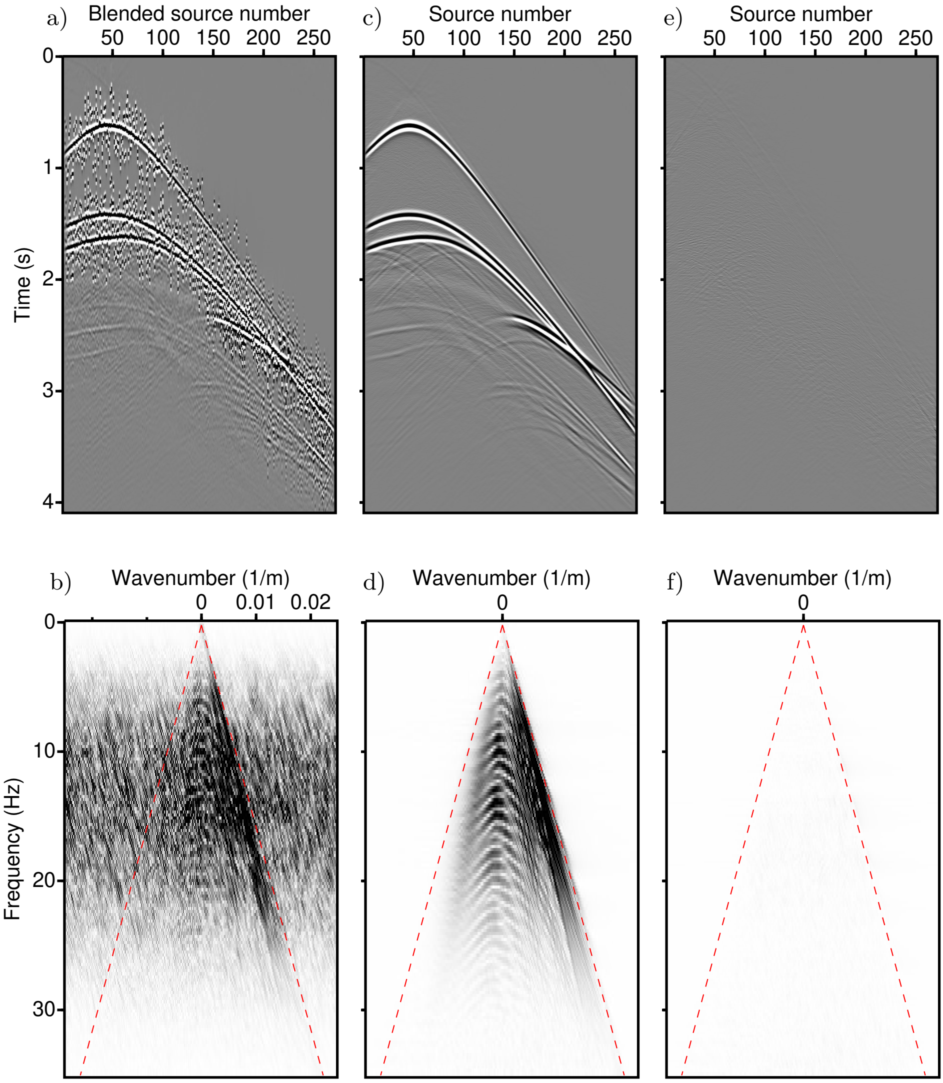


Figure 3.19: *Deblending experiment when the direct wave is filtered out: blended synthetic data in the a) TX domain, b) FK domain; the deblended data in the c) TX domain, d) FK domain; the difference of the deblended and unblended data in the e) TX domain, f) FK domain. As the direct wave is absent, so is the blending noise leakage associated with it.*

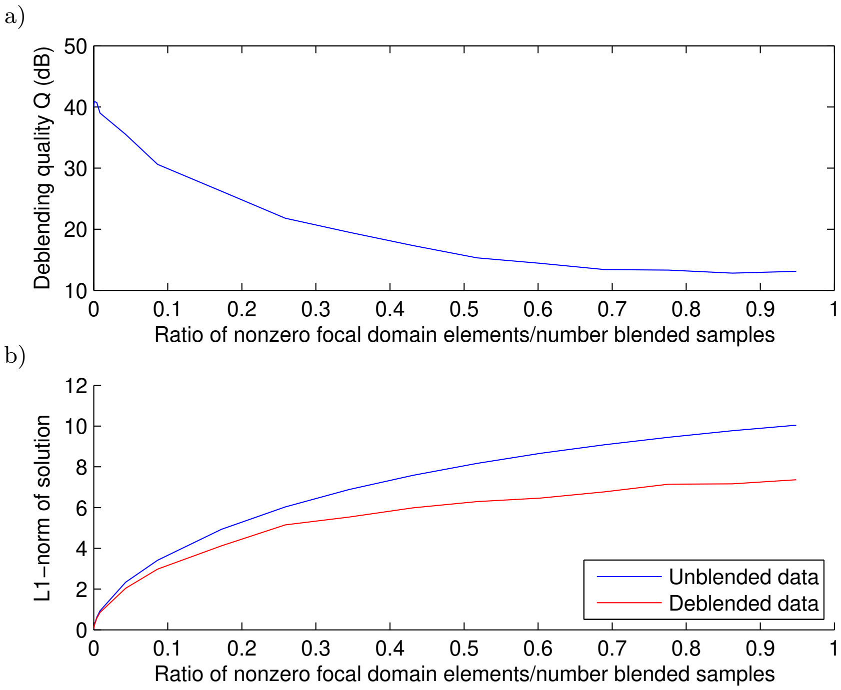


Figure 3.20: *a) Deblending quality versus decreasing sparsity, b) comparison of the ℓ_1-norm of the deblended focal domain against the ℓ_1-norm of the ideal result*

A question that arises naturally then, is under which conditions can we expect exact recovery of unblended data, when using basis pursuit methods. This problem has been studied extensively and multiple criteria for exact recovery have been proposed. They link certain properties of $\mathbf{A}$ to the maximum number of nonzero elements a solution $\mathbf{x}$ can have (i.e. $||\mathbf{x}||_0$) and still be perfectly recovered. Examples of such properties are matrix coherence [Donoho and Elad, 2003; Gribonval et al., 2008], the restricted isometry property [Candès and Tao, 2005, 2006] and the nullspace property [Cohen et al., 2009]. Using such criteria to provide perfect separation guarantees for focal deblending is, unfortunately, not a very fruitful approach. To demonstrate this, I focus on the low matrix coherency criterion for exact recovery.

The coherence $\mu(\mathbf{A})$ of a matrix $\mathbf{A}$ is defined as follows [Donoho and Elad, 2003; Tropp, 2004]:

$$\mu(\mathbf{A}) = \max_{1\leq i\neq j\leq N} \left\{ \frac{|\mathbf{a}_i^{\mathrm{H}}\mathbf{a}_j|}{||\mathbf{a}_i||_2\,||\mathbf{a}_j||_2} \right\}. \tag{3.6.55}$$

In other words, it is the maximum off-diagonal element of $|\bar{\mathbf{A}}^{\mathrm{H}}\bar{\mathbf{A}}|$, where $\bar{\mathbf{A}}$ is $\mathbf{A}$

with its columns scaled such that they have unit ℓ_2-norm. The coherence criterion for exact recovery is given by [Donoho and Elad, 2003; Tropp, 2004]

$$||\mathbf{x}||_0 < \frac{1}{2}\left(1 + \mu(\mathbf{A})^{-1}\right). \tag{3.6.56}$$

As long as the representation of the ideal unblended data in the focal domain satisfies (3.6.56), it is a unique solution and basis pursuit should find it. Matrices $\mathbf{A}$ with lower coherence are preferable, as they allow for exact recovery of less sparse solutions. It should be noted that (3.6.56) guarantees recovery of any n-sparse solution ($n = ||\mathbf{x}||_0$), without imposing any particular structure on nonzero elements of $\mathbf{x}$.

Calculating $\mu(\mathbf{A})$ for focal deblending can be a daunting task as it entails calculating the inner product of all unique combinations of two atoms, from a dictionary of millions of atoms. For our purposes it suffices to examine a small subset of atoms. I extract a two sets of atoms. The first atom of the first set is generated by setting a focal subdomain equal to a Kronecker delta function, taking the inverse focal transform and blending the result. For the rest of the atoms of the first set, the Kronecker delta is shifted in time in steps of $\delta t = 4$ ms and the same process is repeated. For the second set of atoms, the Kronecker delta is shifted in space instead, in steps of $\delta x = 2.5$ m. Note that these shifts are related to the focal domain sampling in time and space and not to blending shifts and source/receiver spacing. In both cases, I calculate the inner product of the first atom of each set with the rest of the elements of the same set. The single-sided focal transform (source side) was used, defined by one focal subdomain with velocity $c = 1489$ m s^{-1} and depth $z = 410$ m. The results can be seen in figure 3.21.

The first value in both plots is equal to 1, as it is the inner product of an atom with itself. These are not taken into account when calculating $\mu(\mathbf{A})$. For the rest of the values, the is a noticeable difference between the time and the space case, the latter tapering off much slower than the first. This implies that the atoms of the space set are closer to being collinear with the first atom of the second set. From a geometrical point of view, it is easy to see why this is the case. Two spatially proximal points in the focal domain generate events with almost coinciding traveltimes, especially around the apex location. These events are then blended, which produces the atoms of the dictionary. These atoms will be almost identical and hence have a large inner product. Two nearby points in time, on the other hand, will lead to atoms that are also shifted in time relative to each other. Owing to the high bandwidth of the generated events, the overlap between two atoms is very little, even for small shifts, leading to low values for the inner product.

The largest value for the inner product of two different atoms in figure 3.21 is 0.93. Plugging into (3.6.56) we get that

$$||\mathbf{x}||_0 < 1.037. \tag{3.6.57}$$

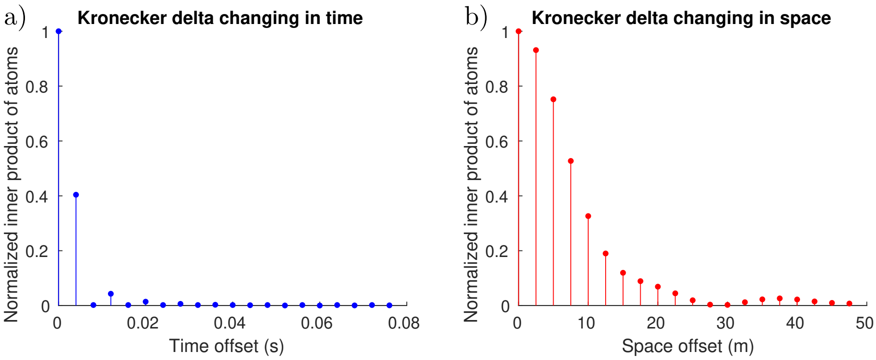


Figure 3.21: *a) Normalized inner products of atoms that correspond to focal domain elements close in time, b) normalized inner products of atoms that correspond to focal domain elements close in space.*

must hold for exact recovery. Essentially this means that there are no practical guarantees regarding exact recovery, due to the high coherence of $\mathbf{A}$ that stems from the physics of wave propagation. That being said, a deblending result can be useful even if the wavefield separation is not perfect and a basis pursuit approach can still be very effective in blending noise removal, as will be seen in the field data examples of chapter 6. It is interesting to note that exact recovery criteria have been used to aid acquisition design, by exploiting the fact that there is often some freedom in designing the measurement matrix $\mathbf{\Psi}$. The design can be optimized to maximize signal recoverability, given a known $\mathbf{\Phi}$. Such an approach was proposed in Campman et al. [2017].

4

Dealing with 3D acquisition using smart subsets

4.1 Motivation

The forward double focal transform, as introduced in chapter 3.3, maps the entire data cube $\mathbf{P}(z_0,\, z_0)$ to its focal representation. In principle, every trace is involved in the calculation. For the ideal, aliasing-noise-free case, the wavefield stored in $\mathbf{P}(z_0,\, z_0)$ should be well-sampled in both source and receiver (x, y)-dimensions, which implies carpet coverage of the acquisition area with sources and receivers. Although having such a densely sampled wavefield available is desirable from a processing point of view, the associated data acquisition costs are prohibitive, at least for exploration purposes. What happens in practice is that one of the source or receiver dimensions is subsampled compared to the other, as a result of a tradeoff between acquisition cost, processing requirements. In some cases terrain difficulties (e.g., presence of offshore platforms), or limitations of the acquisition setup (e.g., maximum cable length in streamer acquisition) place limits on how the wavefield is to be sampled. In cases of severe undersampling of the source or receiver dimension, one of the single-sided variants of the focal transform may be better suited for deblending.

In this chapter I investigate the idea of deblending independently subsets of the 3D data. The subsets and type of focal transform are tailored to the type of acquisition. Here I discuss the cases of towed streamer [Kontakis and Verschuur, 2016] and OBN acquisition [Kontakis and Verschuur, 2017a]. The synthetic data used in the examples were produced using 3D acoustic finite-difference modeling on a 5 km×5 km×2.5 km volume extracted from the SEAM overthrust model, on which a 300 m water layer was prepended. The density model was generated from the velocity model using Gardner's relation. Two slices of the velocity model are shown in figure 4.1.

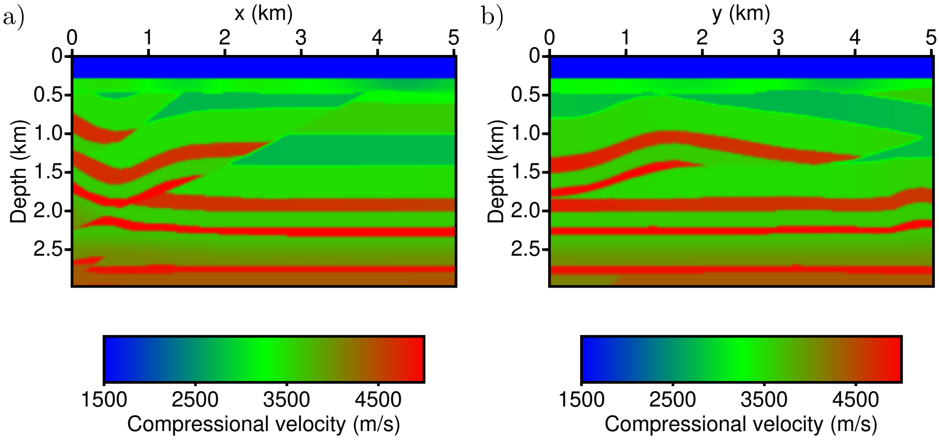


Figure 4.1: *Two slices of the compressional velocity model used for the synthetic examples shown in this chapter. a) A y-slice of the velocity model at* $y = 2.7$ *km; b) an x-slice of the velocity model at* $x = 2$ *km.*

4.2 The streamer case

Multi-streamer acquisitions, such as the one shown in figure 4.2, are 3D and are characterized by a combination of relatively a well-sampled inline direction and a more coarsely-sampled crossline direction. Depending on how deblending is performed, the crossline dimension may require on-the-fly interpolation to combat undersampling. Even when interpolation can be avoided, processing all the data jointly can be a considerable computational burden, given the amount of convolution/correlation operations involved in carrying out the focusing/defocusing steps. Fortunately, we can take advantage of good inline sampling, by treating the full 3D problem as a set of pseudo-2D problems, each involving a single streamer and multiple vessels shooting in a blended fashion.

This modification converts the problem into a pseudo-2D problem, in which every streamer is handled independently. The receivers on a particular streamer along with the shot locations define a set of midpoints that can be used to construct a set of focal grid points. For the acquisition in figure 4.2, two separate grids are

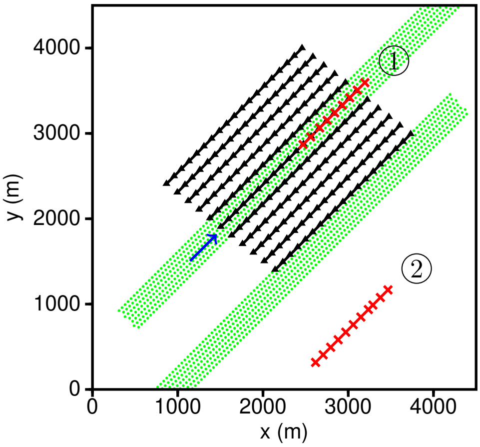


Figure 4.2: *Wide azimuth streamer acquisition. The red x symbols represent shots carried out by two vessels, marked 1 and 2, respectively. Streamers are depicted as black lines. The streamer indicated by the blue arrow together with the shots from the two vessels forms the subset that will be deblended. The points in green define two focal grids.*

used, each with their own associated focal subdomains and focal operators, centered around the subset midpoints. The purpose of each focal subdomain is to explain the wavefield created by each of the sources. Confining the grids to lie around midpoints reduces the number of arithmetic operations needed for applying the focal operators, but comes at the cost of reduced focusing for reflections that do not originate from locations close to the midpoints at each chosen depth level. This happens because the moveout of the focusing operators will not match as closely the moveout of events away from the focusing points, due to cross-line variations in the medium.

In general, when using n_{gr} grids, the deblending problem for each of the streamer subsets becomes

$$\min_{\substack{\hat{\mathbf{X}}^D_{k,j}(z_k,\,z_k)\\ k=1,\ldots,K\\ j=1,\ldots,n_{\mathrm{gr}}}} \left\{ \sum_t \sum_{j=1}^{n_{\mathrm{gr}}} \sum_{k=1}^{K} \left|\left| \hat{\mathbf{X}}^D_{k,j}(z_k,\, z_k) \right|\right|_{\mathrm{S}} \right\} \quad \text{subject to}$$

$$\sum_\omega \left|\left| \mathbf{P}^{\mathrm{str}}_{\mathrm{bl}}(z_0,\, z_0) - \left[\mathbf{P}_1(z_0,\, z_0),\ \ldots,\ \mathbf{P}_{n_{\mathrm{gr}}}(z_0,\, z_0)\right] \boldsymbol{\Gamma}^{\mathrm{str}}(z_0) \right|\right|_{\mathrm{F}} \leq \sigma, \tag{4.2.1}$$

with

$$\mathbf{P}_j(z_0,\, z_0) = \sum_{k=1}^{K} \mathbf{W}^-_{k,j}(z_0,\, z_k) \mathbf{X}^D_{k,j}(z_k,\, z_j) \mathbf{W}^+_{k,j}(z_k,\, z_0). \tag{4.2.2}$$

Here $\mathbf{P}_{\mathrm{bl}}^{\mathrm{str}}(z_0,\, z_0)$ is the blended data recorded by the selected streamer and $\mathbf{\Gamma}^{\mathrm{str}}(z_0)$ the part of the blending operator that produces it from unblended data. $\mathbf{P}_j(z_0,\, z_0)$ is the data synthesized by defocusing the focal subdomains associated with the j-th grid, $\mathbf{X}_{k,j}^{D}(z_k,\, z_j)$, $k = 1, 2, \ldots, K$, followed by stacking the defocused data cubes. The j-th grid also has related focal operators $\mathbf{W}_{k,j}^{-/+}(z_0,\, z_k)$ for each of the K depth levels.

For the example in figure 4.2, the sail line consists of 11 streamers, with a crossline spacing of 100 m. Each streamer has 50 receivers with 25 m inline spacing, for a total streamer length of 1.25 km. A total of 50 blended shot gathers are acquired, approximately every 25 m. The airguns fire with random time delays ranging from 0 s to 0.5 s, in a blended fashion. Three common receiver gathers of unblended and pseudo-deblended data can be seen in figure 4.3a and 4.3b, respectively.

Deblending is done using the two focal grids depicted in figure 4.2. The spacing used is 25 m along both the inline and crossline axes. Each grid covers an area of approximately 300 m×5350 m. Local subsurface offsets extend from -3 km to +3 km along the axis parallel to the streamers. Seven focal operators were constructed with velocities ranging from 1493 m s^{-1} to 1621 m s^{-1}, and depths ranging from 300 m to 2023 m. The number maximum number of SPGL1 iterations was set to 1000. Three deblended common receiver gathers and the difference from the unblended data can be seen in figures 4.4a and 4.4b, respectively. The gathers correspond to those of figure 4.3. Traces from the first, third and fifth focal subdomain are shown in figures 4.5a-c. While some amount of compression can be observed, it is likely that the focal grids do not have enough extent, or are not optimally placed, to be able to adequately explain the strong seabed multiples that dominate the input data. This can then lead to the increased amount of blending-noise leakage seen in this result.

Since in this particular case the residual was nonzero, it was pseudo-deblended and added back to the deblended result,

$$\mathbf{P}_{\mathrm{debl+res}}^{\mathrm{str}}(z_0,\, z_0) = \mathbf{P}_{\mathrm{debl}}^{\mathrm{str}}(z_0,\, z_0) + \mathbf{P}_{\mathrm{res}}^{\mathrm{str}}(z_0,\, z_0)\mathbf{\Gamma}^{\mathrm{str}}(z_0)^{\mathrm{H}}, \tag{4.2.3}$$

where

$$\mathbf{P}_{\mathrm{res}}^{\mathrm{str}}(z_0,\, z_0) = \mathbf{P}_{\mathrm{bl}}^{\mathrm{str}}(z_0,\, z_0) - \mathbf{P}_{\mathrm{debl}}^{\mathrm{str}}(z_0,\, z_0)\mathbf{\Gamma}^{\mathrm{str}}(z_0) \tag{4.2.4}$$

is the residual. This is a commonly used method to add back coherent energy that may have remained in the residual, usually at the cost of also increasing the amount of blending noise in the result. This can be partially mitigated by also doing some extra denoising before adding back. For the result shown here, the signal-to-blending noise ratio after deblending and adding back the residual becomes 17.9dB.

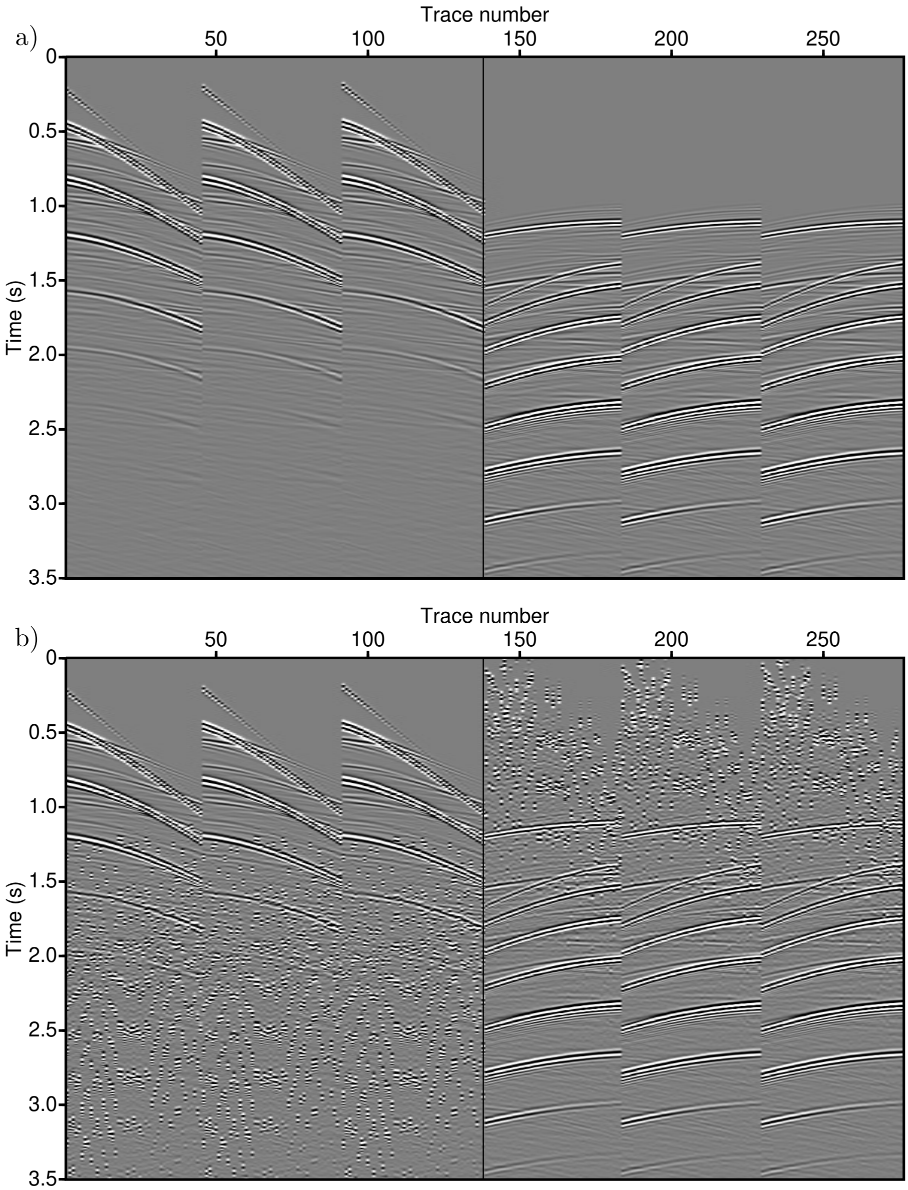


Figure 4.3: *Three common receiver gathers showing a) unblended and b) pseudo-deblended data. The shots originate from the airgun of vessel 1 (left side) and vessel 2 (right side), respectively.*

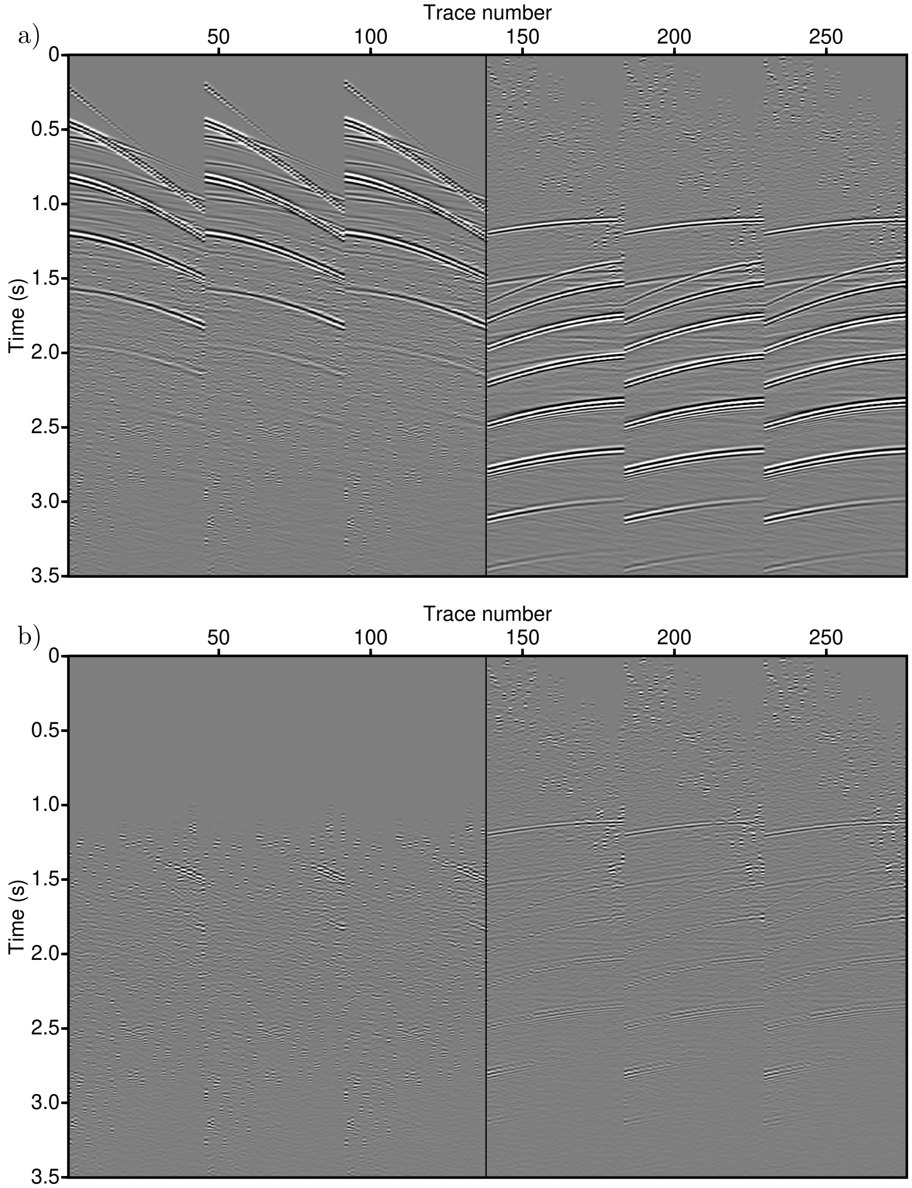


Figure 4.4: *Three common receiver gathers showing a) deblended data, b) difference of deblended from unblended data. The shots originate from the airgun of vessel 1 (left side) and vessel 2 (right side), respectively.*

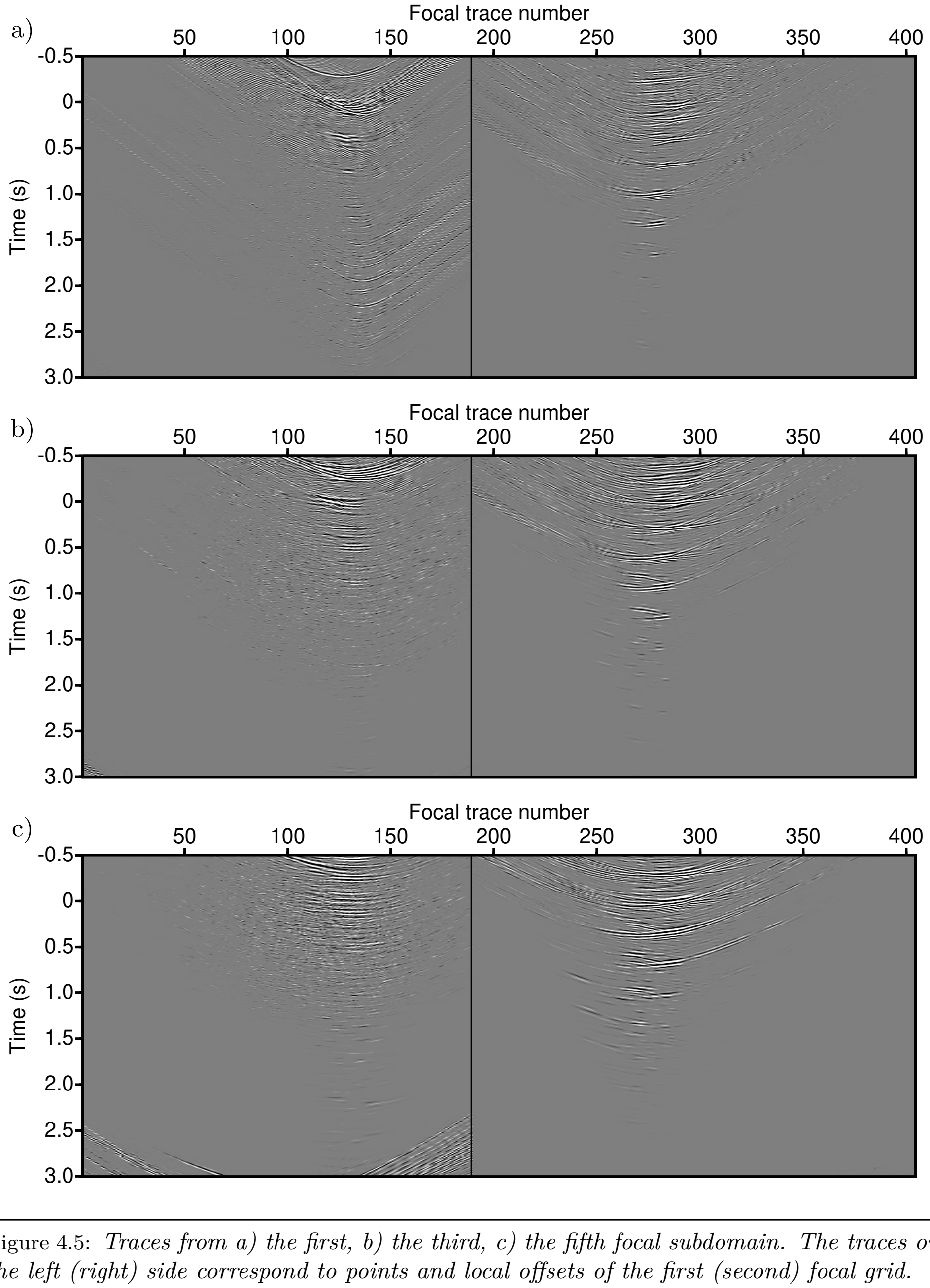


Figure 4.5: *Traces from a) the first, b) the third, c) the fifth focal subdomain. The traces on the left (right) side correspond to points and local offsets of the first (second) focal grid.*

4.3 The OBN case

In OBN acquisitions the few receiver nodes are installed at the seabed, usually in a sparse arrangement. One or more sources are towed by a vessel and are activated at regular intervals. Source sampling is typically much denser than receiver sampling. The relatively small number of receivers means that when applying focusing using the double focal transform, the receiver dimension does not contribute significantly, and will produce strong aliasing artifacts. Instead of applying a version of focal deblending that operates on the entire dataset, an alternative idea is to deblend each common receiver gather independently and use the single-sided focal transform (3.4.24). Parallelization with respect to receiver gathers becomes then trivial and a smaller problem has to be solved for each gather. No modification to the blending operator is needed, since it acts on independent receiver gathers. The deblending problem for each individual receiver at location $\mathbf{r}$ becomes

$$\min_{\substack{\hat{\mathbf{X}}_k^S(z_0)\\ k=1,\dots,K}} \left\{ \sum_t \sum_{k=1}^{K} \left|\left| \hat{\mathbf{X}}_k^S(z_0) \right|\right|_{\mathrm{S}} \right\} \quad \text{subject to}$$

$$\sum_\omega \left|\left| \mathbf{p}_\mathbf{r}(z_{\mathrm{sb}},\, z_0) - \sum_{k=1}^{K} \mathbf{w}^{\pm}_{\mathbf{r},k}(z_{\mathrm{sb}},\, z_0)\mathbf{X}_k^S(z_0)\mathbf{\Gamma}(z_0) \right|\right|_{\mathrm{F}} \leq \sigma. \tag{4.3.5}$$

Here z_{sb} is the depth of the seabed, $\mathbf{p}_\mathbf{r}(z_{\mathrm{sb}},\, z_0)$ is the common receiver gather for the receiver at $\mathbf{r}$ (i.e. a row vector) and $\mathbf{w}^{\pm}_{\mathbf{r},k}(z_{\mathrm{sb}},\, z_0)$ (also a row vector) is a two-way propagator that generates the upgoing wavefield recorded by the receiver, after reflection of the downgoing wavefield (produced by each virtual source in $\mathbf{X}_k^S(z_0)$) by flat reflector at z_k. The lowercase forms $\mathbf{p}_\mathbf{r}(z_{\mathrm{sb}},\, z_0)$ and $\mathbf{w}^{\pm}_{\mathbf{r},k}(z_{\mathrm{sb}},\, z_0)$ are used here to highlight that the data misfit is over a single common receiver gather.

For this example, a racetrack-type OBN acquisition is simulated using the model described in chapter 4.1. The scenario that is simulated is that of two vessels that carry sources that are activated independently. The trajectories of the two vessels and the shot locations are depicted in figure 4.6. Each boat moves at an average speed of 2.5 m s^{-1} and fires a shot every 10 s with an additional random time delay between 0 s and 2 s. The average spacing between shots is 27.5 m and the inter-racetrack spacing is approximately 25 m. Using two boats may increase the overall cost of the acquisition but it speeds it up by a factor of 2, which may be desirable when there is a very small time window for the acquisition to take place, e.g., when an area is very busy or when weather conditions are not favorable.

Figure 4.7a shows 500 traces of common receiver data modeled for the receiver at (2km, 2.7km), marked by a green inverted triangle in figure 4.6. The corresponding pseudo-deblended data is shown in figure 4.7b.

For the deblending procedure a focal grid must be defined. For this example, a rectangular grid is used, with a spacing of 25 m both in the x- and y-dimensions. The

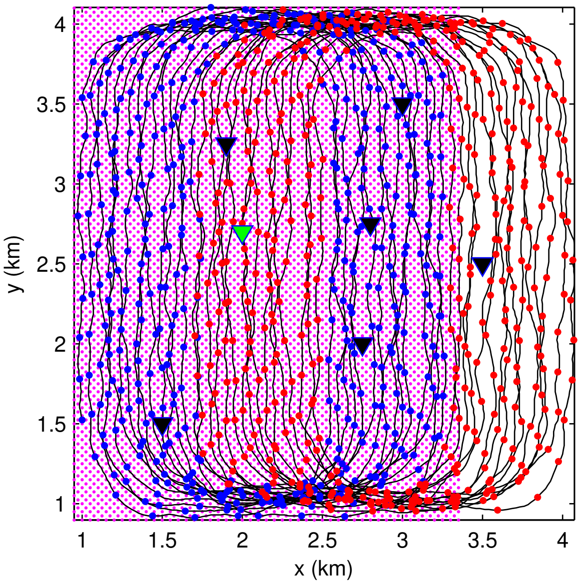


Figure 4.6: *Racetrack acquisition at the surface. Blue (red) dots represent shots carried out by the first (second) boat. Inverted triangles denote receiver nodes. The green triangle defines the data subset to be deblended. The magenta points denote the focal grid.*

grid covers the acquisition area covered by the first vessel, which is approximately 3.4 km×4 km. Four focal operators are used, with velocities in the range from 1500 m s^{-1} to 2250 m s^{-1} and depths from 450 m to 1000 m. These were designed with the purpose of approximately focusing the seabed reflection and the strongest associated surface-related multiples. Figure 4.9 shows part of the focal domain content. Using the single-sided focal transform changes the nature of the focal domain, which then functions more as a collection of source arrays.

The deblending result and the corresponding difference from the unblended data are shown in figures 4.8a and 4.8b, respectively. The incoherence of the blending noise in common receiver gathers allows the suppression of most of the blending noise, even when treating receivers separately. The separation is not perfect however; this is likely a combination of insufficient focusing and the focal operator range limitations of the focal operators discussed in chapter 3.6. Insufficient focusing, in particular, can be the result of the approximate velocity model used, but also of the fact that in this example only a subset of the data contributes in the focusing calculations. The signal-to-blending-noise ratio after deblending for this example is 20dB.

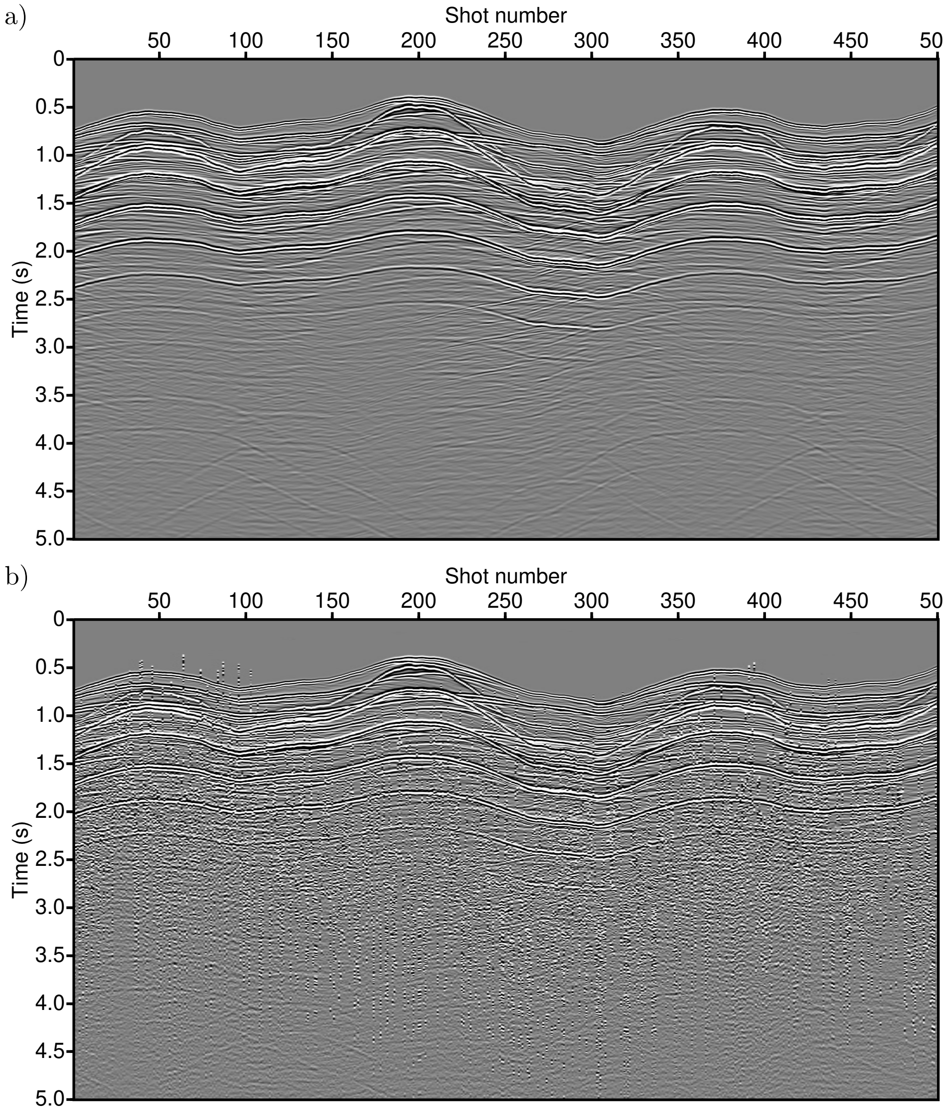


Figure 4.7: *a) Data modeled at the receiver location marked in green in figure 4.6, b) the corresponding pseudo-deblended data.*

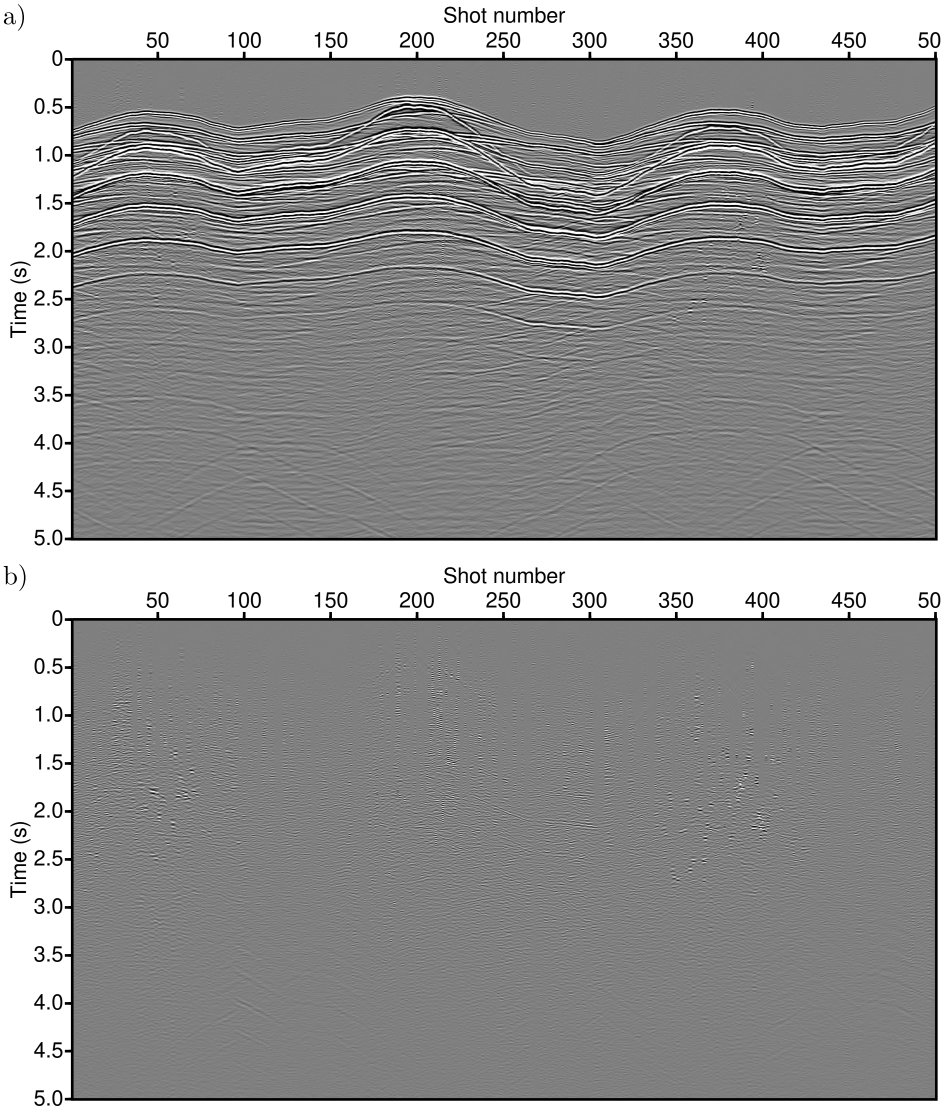


Figure 4.8: *a) Deblended data, b) difference from corresponding unblended data.*

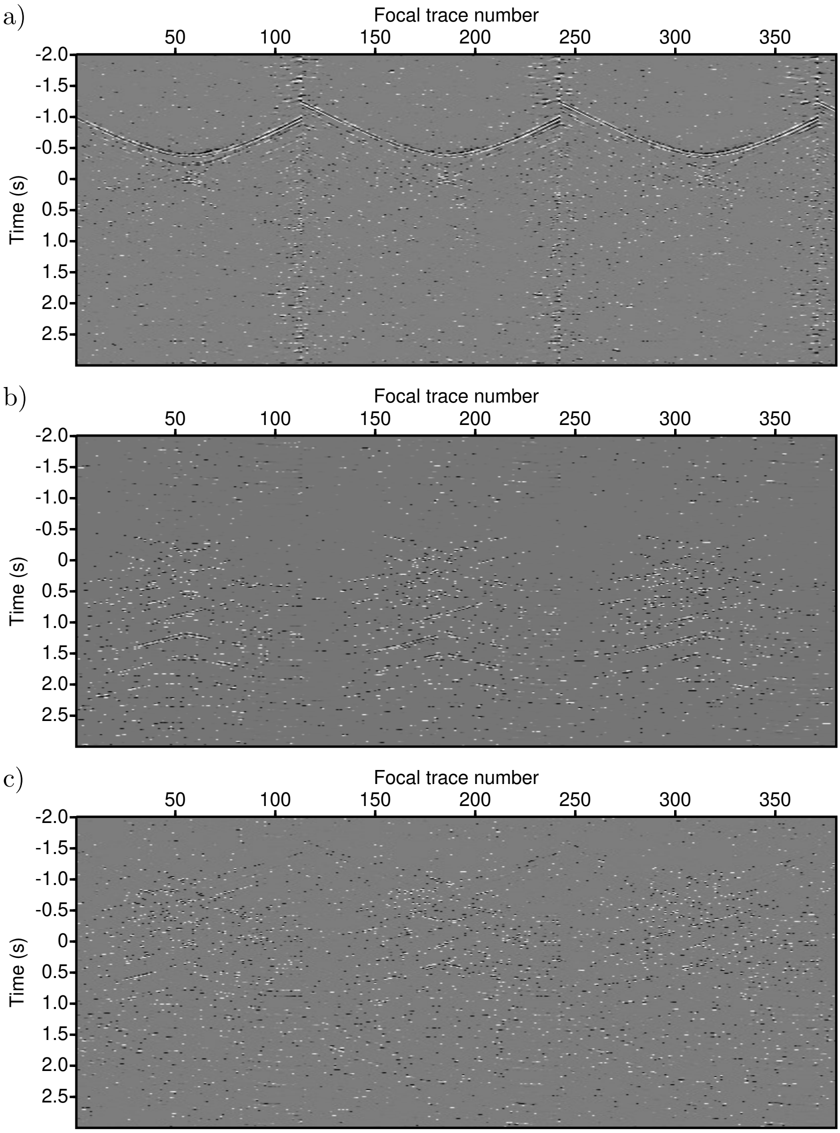


Figure 4.9: *Traces from a) the first, b) the second, c) the fourth focal subdomain. Note that the focal subdomains here exhibit a different character compared to those of from the streamer example. This is due to the fact that a different kind of focal transform was used for this example.*

5

Hybrid deblending solutions

5.1 Motivation

In section 3.6.1 I identified a potential challenge for focal deblending. Focal operators are linear operators that in practice have a limited span. As I demonstrated in section 3.6.1, this can lead to additional blending noise leakage into the deblended wavefields. In this chapter I investigate a possible remedy for this problem, which is to combine the focal transform with the fast discrete curvelet transform [Kontakis and Verschuur, 2017b]. A similar approach using the linear Radon transform can be found in Kontakis and Verschuur [2015]. The crucially important property of the particular flavor of curvelet transform used here is that, unlike the focal transform, it is a unitary isometry. Although this property alone does not guarantee perfect deblending, it at least guarantees that the whole wavefield can be explained. The two transforms have atoms with different shapes and act in a complementary fashion. As the atoms of the focal transform have a hyperbolic shape, describing a linear event, such as a direct wave, needs a superposition of many individual atoms. Curvelets, on the other hand, can describe such events more efficiently. The advantages and disadvantages of a hybrid approach are further discussed in this chapter.

5.2 The curvelet transform

The curvelet transform borrows from multiscale analysis the idea of representing a signal as a combination of features that manifest themselves at different scales. Its second major characteristic is that most curvelet elements have have a well-defined orientation in 2D or 3D space. They were first introduced by Candes and Donoho

[2000], and since then a family of variants and different numerical implementations has emerged. The variant used here for the focal-curvelet hybrid deblending algorithm is the fast discrete curvelet transform (FDCT) through frequency wrapping, which is the second implementation discussed by Candès et al. [2006]. The authors mention a list of properties of the transform, of which the following are of particular interest for our application:

- The frequency wrapping-based FDCT is a unitary transform, implying that the adjoint transform is also the inverse. Loosely speaking, there is no 'information loss' when transforming to and from the curvelet domain, which is not guaranteed in the case of the focal transform.
- Square-integrable functions over $\mathbf{R}^2$ can be expanded as a series of curvelets. In our case the role of the function is played by a common receiver gather where the first dimension is time and the second dimension is space. This property is used to remedy the range difficulties of the focal transform.
- Curvelets are very efficient in explaining shapes with edges, which makes them a good candidate for seismic data processing and compression. This is due to the shape of the individual elements, which are localized both in space and frequency. This can be seen in the examples shown in figure 5.1. Except for the coarsest scale, atoms have a specific orientation.
- Curvelets are well-suited for solving certain very ill-posed problems. Candès et al. [2006] mention medical imaging applications, however since then a plethora of publications have established its efficacy in seismic processing applications as well, especially for interpolation and deblending problems. Several examples are mentioned below.

Curvelets were quickly adopted for seismic data processing and have been used in almost every major step of the processing worflow. Examples include denoising [Candès et al., 2006], groundroll attenuation [Naghizadeh and Sacchi, 2018], regularization/interpolation [Hennenfent et al., 2010], deblending [Kumar et al., 2015; Zu et al., 2016b], migration [Douma and de Hoop, 2007; Chauris and Nguyen, 2008], and surface multiple estimation/removal [Herrmann et al., 2008; Donno et al., 2010].

5.3 The hybrid focal-curvelet transform

A relatively straightforward way to extend focal deblending to its focal-curvelet hybrid form is to redefine the deblending optimization problem 3.5.36 as

$$\mathbf{x}_{fc,\mathrm{debl}} = \underset{\mathbf{x}_{fc}}{\arg\min} \left\{ ||\mathbf{x}_{fc}||_1 \right\} \quad \text{subject to} \quad ||\mathbf{p}_{\mathrm{bl}} - \boldsymbol{\Psi}\boldsymbol{\Phi}_{fc}\mathbf{x}_{fc}||_2 \leq \sigma, \tag{5.3.1}$$

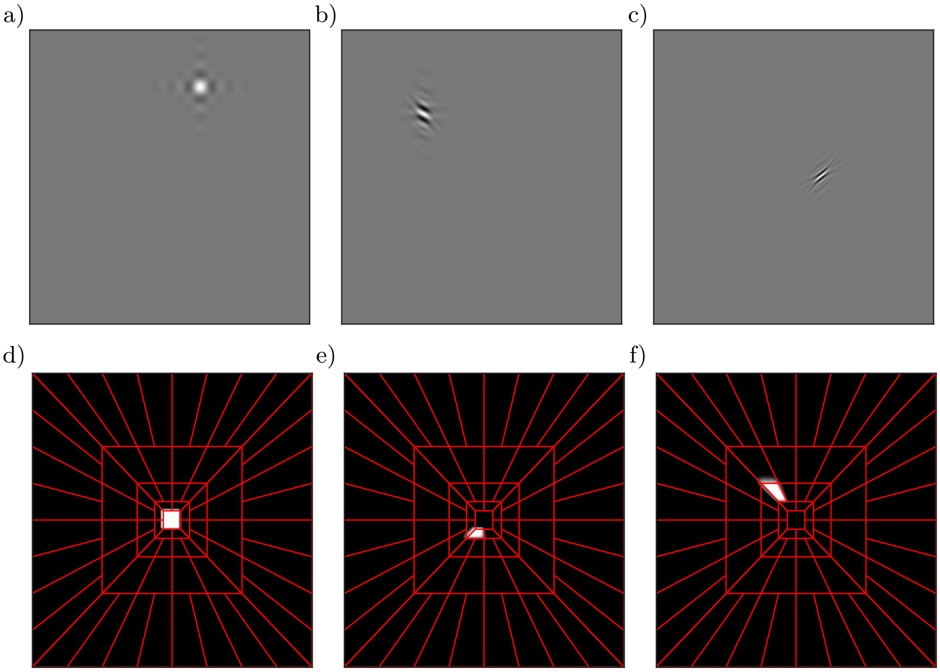


Figure 5.1: *Three curvelet elements in the TX and FK domains: a) first scale, TX domain; b) second scale, sixth orientation, TX domain; c) third scale, first orientation, TX domain; d-f) FK transform of the curvelet elements related to a-c). The red lines demarcate the area occupied by each curvelet element.*

with

$$\mathbf{\Phi}_{fc} = [\mathbf{\Phi}_f \,|\, \mathbf{\Phi}_c] \quad \text{and} \quad \mathbf{x}_{fc} = \begin{bmatrix} \mathbf{x}_f \\ \mathbf{x}_c \end{bmatrix}. \tag{5.3.2}$$

Here I have again used the vectorized form of the problem to keep the mathematical expressions concise. $\mathbf{\Phi}_f$ and $\mathbf{\Phi}_c$ are linear operators representing the focal (for all depths) and the curvelet transform, respectively. The curvelet transform used is the 2D version applied onto common receiver gathers, to take advantage of blending noise incoherence. The vectors $\mathbf{x}_f$ and $\mathbf{x}_c$ hold the corresponding transform domain coefficients. The difference between this and formulation (3.5.36) from chapter 3.5 is the dictionary: $\mathbf{\Phi}_{fc}$ is formed by concatenating the atoms of two different dictionaries. This approach has some advantages:

- We now have an extended dictionary, with a subset of its atoms forming a basis.

- Because there is a greater variety of atoms with different 'shapes', features of the signal that would need a linear combination of many atoms of one of the transforms, may sometimes be explained using only a few atoms of the extended dictionary, leading to a sparser representation.

It is, however, a solution with its own challenges:

- The overall coherence of the total dictionary goes up, as we are concatenating the atoms of the curvelet dictionary, which on their own form a basis, with the focal dictionary. This increases the sparsity requirements for guaranteed prefect recovery (see discussion in section 3.6.1).

- The relative scaling of the atoms becomes important. If the atoms are not normalized, one of the two transforms might end up having more high amplitude components in $\mathbf{x}_{fc}$, purely due to scaling reasons. This may leave the other transform contributing very little to the solution. A possible remedy is to ensure that all atoms have unit norm by prescaling them:

  $$\mathbf{\Phi}_{fc} = \left[\mathbf{\Phi}_f \mathbf{\Lambda}_f \,|\, \lambda_c \mathbf{\Phi}_c \mathbf{\Lambda}_c\right], \tag{5.3.3}$$

  where $\mathbf{\Lambda}_f$ and $\mathbf{\Lambda}_c$ are diagonal matrices that scale the atoms of the two transforms to unit norm. The tuning parameter λ_c controls the overall 'importance' of the curvelet transform relative to the focal transform in the optimization. In cases where the dictionaries are given in explicit matrix form, this scaling needs to be applied only once to the matrix as a preparation step. However, if the transforms are implemented as matrix-free operators, this scaling has to be applied as a separate operation every time the forward or adjoint are calculated. Both transforms in our case have matrix-free implementations.

- The computational cost increases as two transforms have to be applied every time the forward or adjoint operation is used in the optimization scheme.

Instead of concatenating two dictionaries, as in (5.3.2), it is possible to compose them, i.e. $\mathbf{\Phi}_{fc} = \mathbf{\Phi}_f \mathbf{\Phi}_c$. In this way the focal representation of the data could be further compressed by the curvelet transform. This approach has two main drawbacks. The first is the increased computational cost, as the curvelet transform would have to be applied on the focal representation of the data, which can be several times the size of the observed data. The second important drawback is that the curvelet transform would not be able to help overcome the range limitations of the focal transform.

Another formulation that could be useful in certain situations is to set

$$\mathbf{\Phi}_{fc} = \left[\mathbf{M}_f \mathbf{\Phi}_f \,|\, \mathbf{M}_c \mathbf{\Phi}_c\right]. \tag{5.3.4}$$

Both $\mathbf{M}_f$ and $\mathbf{M}_c$ are diagonal matrices that act as windows that split the seismic data in separate regions, possibly with an overlap to reduce edge effects. Each transform then is tasked with explaining a different part of the data, e.g. the focal transform could handle the near-offset part and leave the far-offset part for curvelets. This can be a way to deal with the scaling issues mentioned previously, but it may reintroduce range-related problems for the part handled by the focal transform, similar to the composition approach mentioned above. In the following examples we have opted for (5.3.2) rather than (5.3.4).

5.4 Examples on synthetic data

We now revisit the example shown in section 3.6.1. The problem there was that the focal operators filtered out a large part of the direct wave. We first examine the reconstruction case, i.e. no blending is performed and $\boldsymbol{\Psi} = \mathbf{I}$ in (5.3.1). For the curvelet transform I used 8 scales and 32 angles for the second scale. The number of angles increases by a factor of 2 per each two higher scales, ending at 256 angles for the eighth scale. Figure 5.2 shows that the hybrid transform is now able to explain the direct wave. This is also clearly seen when comparing FK spectra (figures 5.3c and 5.3e).

While the reconstruction example is promising in that it shows that the full dataset can be now explained, the more important question is whether this capability leads to a better deblending result. Another thing to consider is whether using the hybrid transform is any better than just using curvelets alone. To investigate this question, I deblend the data shown in figure 3.18 using i) only the focal transform, ii) the hybrid focal-curvelet transform with $\lambda_c = 0.07$ and iii) only the curvelet transform. The results are shown in figure 5.4. Although not perfect, the results of the hybrid transform are noticeably better than using any of the transforms alone. The focal transform seems to handle the more curved near-offset part better, whereas curvelets handle better the far-offset part, where the moveout of the seismic events is more linear. It also handles the direct wave. This is possible due to the shape of the curvelet atoms, which resemble small packets of linear-moveout events translated and rotated (here) in a 2D space. This example demonstrates that dealing with range-related limitations can affect significantly the deblending quality.

The relative scaling between the two transforms has an effect on how much each transform contributes to the final deblended solution, but also on how blending-noise leakage manifests itself in the end result. Comparing figures 5.4c and 5.4g it is evident that each of the two transforms leads to blending-noise leakage occuring at different locations. In figure 5.5 the time-domain contribution of each of the two transforms is illustrated, as well as the difference between the deblended and the unblended gather, for three different values of the relative scaling parameter λ_c. The total deblended result is the sum of the contributions of each transform.

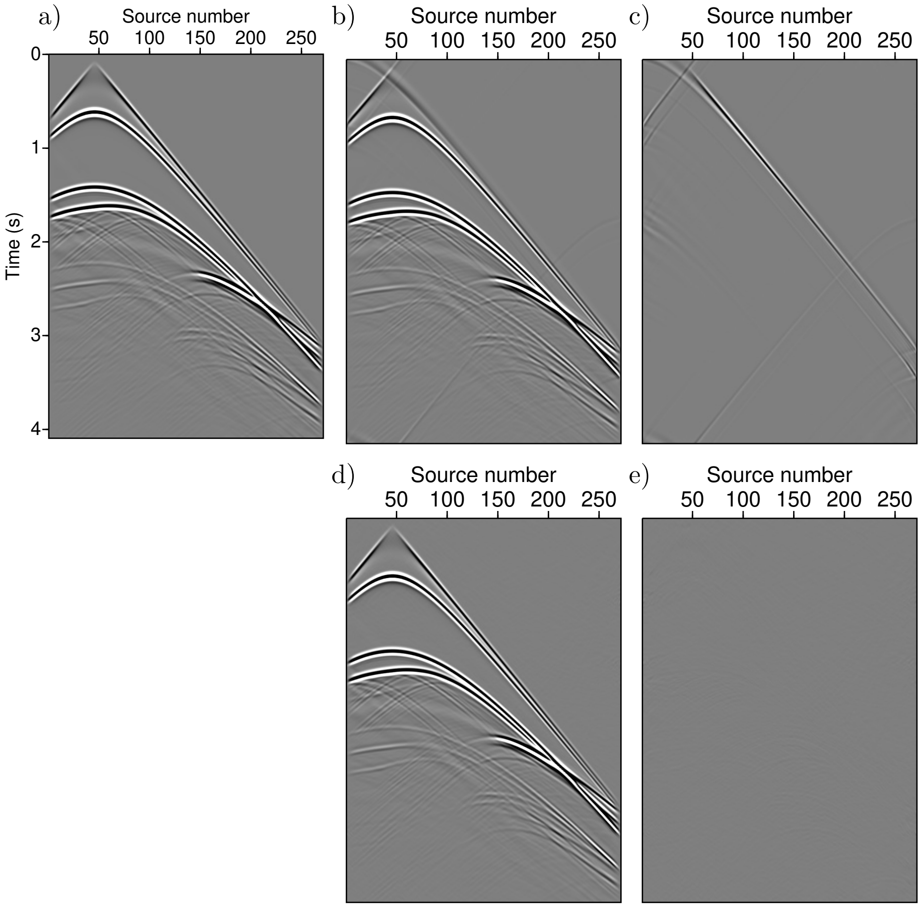


Figure 5.2: *Reconstruction comparison (TX domain): a) the original synthetic dataset, b) the reconstructed dataset (focal transform only), c) difference between reconstructed and original dataset (focal transform only), d) the reconstructed dataset (hybrid transform), e) difference between reconstructed and original dataset (hybrid transform).*

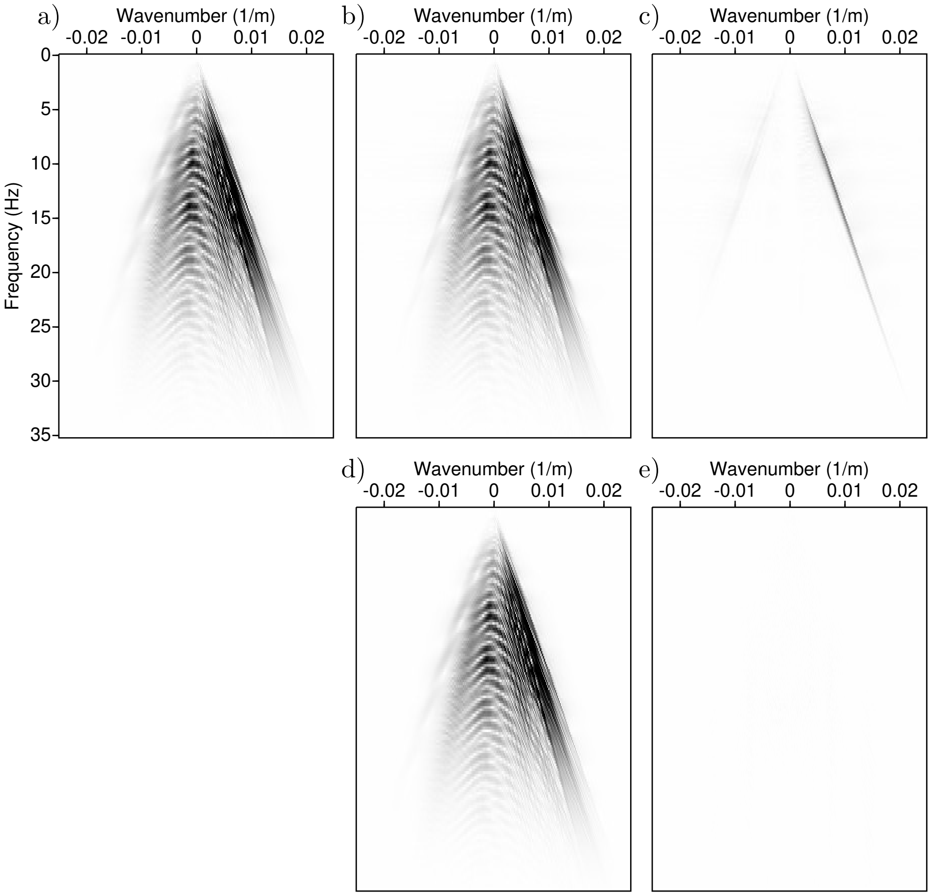


Figure 5.3: *Reconstruction comparison (FK domain): a) the original synthetic dataset, b) the reconstructed dataset (focal transform only), c) difference between reconstructed and original dataset (focal transform only), d) the reconstructed dataset (hybrid transform), e) difference between reconstructed and original dataset (hybrid transform).*

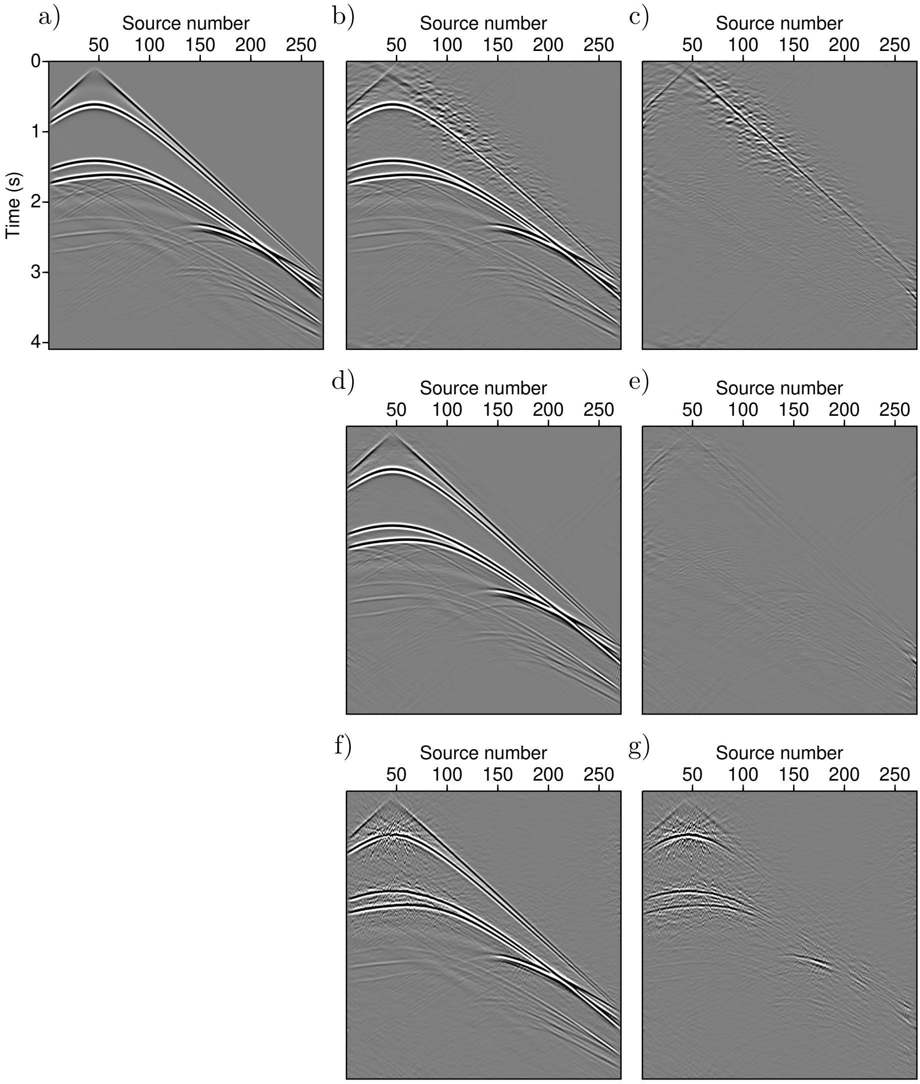


Figure 5.4: *Deblending comparison (TX domain): a) unblended gather, b) deblended gather (focal transform only), c) difference between the unblended and the deblended gather (focal transform only), d) deblended gather (hybrid transform), e) difference between the unblended and the deblended gather (hybrid transform), f) deblended gather (curvelet transform only), g) difference between the unblended and the deblended gather (curvelet transform only).*

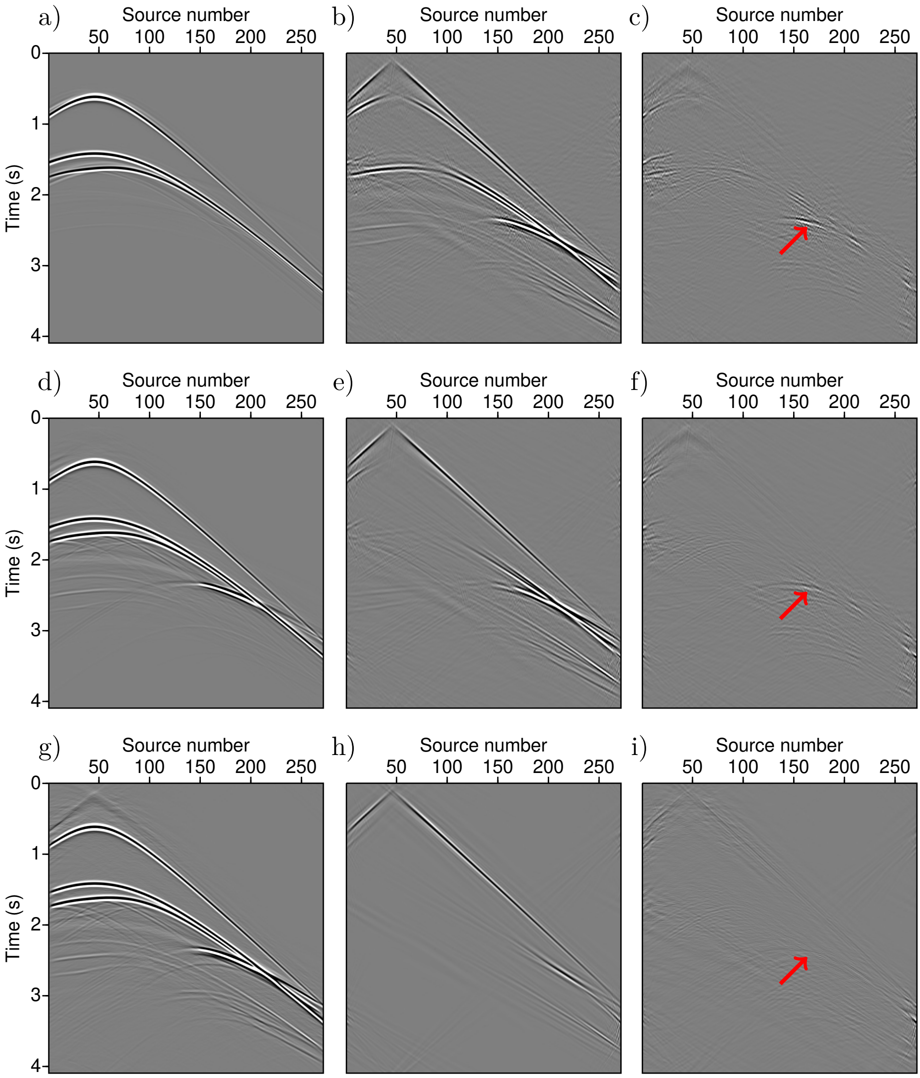


Figure 5.5: *Effect of the relative scaling* λ_c *on transform contribution balance and noise leakage: a) Focal contribution for* $\lambda_c = 1$*; b) Curvelet contribution for* $\lambda_c = 1$*; c) Difference from unblended data for* $\lambda_c = 1$*; d) Focal contribution for* $\lambda_c = 0.33$*; e) Curvelet contribution for* $\lambda_c = 0.33$*; f) Difference from unblended data for* $\lambda_c = 0.33$*; g) Focal contribution for* $\lambda_c = 0.07$*; h) Curvelet contribution for* $\lambda_c = 0.07$*; i) Difference from unblended data for* $\lambda_c = 0.07$*. The red arrow points at curvelet-associated blending noise leakage.*

As λ_c becomes progressively smaller, the curvelet transform is further restricted to explaining mostly linear events using few atoms. The blending-noise leakage pointed at by the red arrow in figures 5.5c, 5.5f and 5.5i enters the solution via the curvelet transform. Reducing λ_c, therefore, will tend to reduce curvelet-associated leakage too. The noise-leakage tradeoff between the transforms can be controlled via λ_c, potentially leading to better-quality results. The difficulty here, however, is that choosing a value for λ_c is not straightforward, as the presence of blending noise will not be always easy to detect in order to judge which value for λ_c is optimal. The blending equation will not be of help, since for $\lambda_c > 0$ each result after convergence will practically satisfy it. For strong, recognizable blending noise remaining in the result, varying λ_c can lead to effective noise attenuation.

6

Field data examples

The main goals of the previous chapters were to describe the focal deblending method as well as the hybrid and smart subset extensions. In this chapter, the methods developed so far are applied on field data. Focal deblending as described in chapter 3 is applied on streamer data provided by Equinor. The smart subset extension is then tested under two different acquisition scenarios. The streamer acquisition case is examined first, on data provided by CGG. The OBN case follows, using data provided by TEEC. The hybrid transform approach is then tested using the CGG dataset. For the last example, focal deblending is applied on streamer data provided by TGS. Unlike the preceding examples, in which the blending was done numerically, the TGS dataset was already recorded in the field with some amount of wavefield overlap.

6.1 Single line 2D data

The first field data example is based on a marine streamer dataset acquired in the North Sea by SAGA Petroleum A.S, currently part of Equinor. The dataset consists of 151 sources and receivers in a fixed-spread configuration, after some preprocessing as explained later, with a source/receiver spacing of 25 m. The maximum offset is 3750 m. The time sampling interval is 8 ms. Some preprocessing had already been performed on the dataset in order to mimic a 2D line, namely Radon interpolation to fill in the near offsets [Kabir and Verschuur, 1995] and data duplication using the reciprocity principle to generate negative offsets. Moreover, surface-related multiples had been already removed using SRME [Verschuur et al., 1992]. Three common shot and common receiver gathers can be seen in figure 6.1.

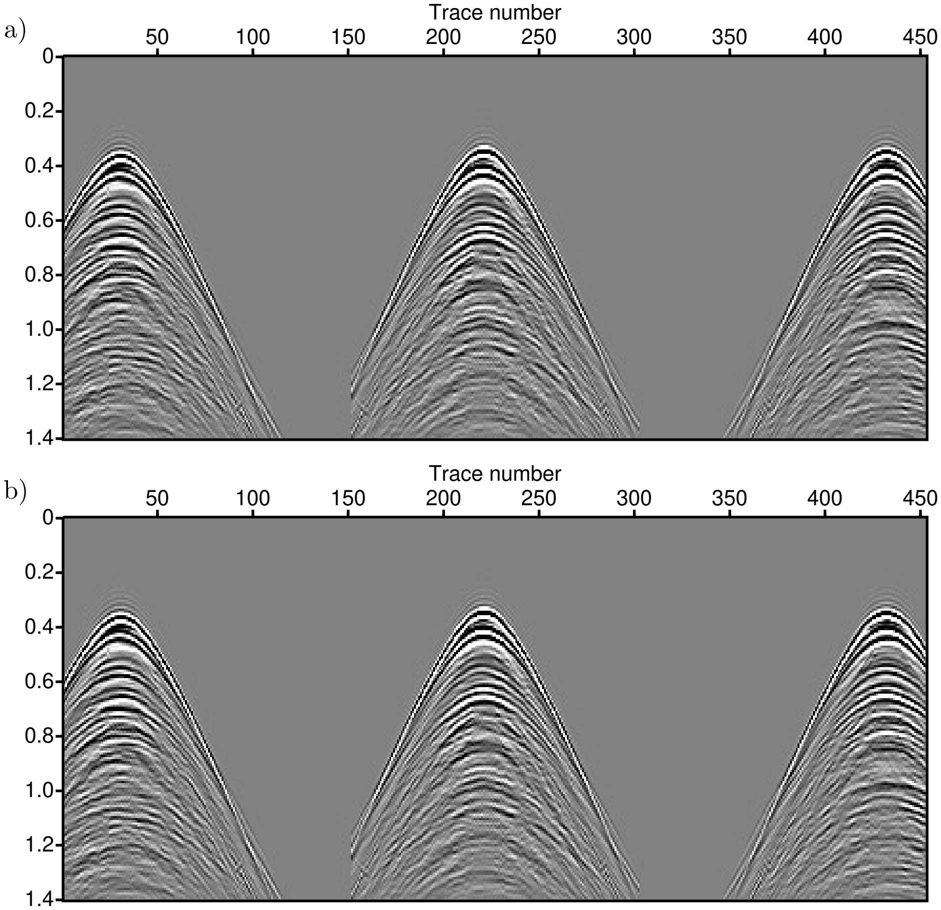


Figure 6.1: *2D line example, unblended a) common shot gathers, b) common receiver gathers. Note that due to the applied reciprocity, the figures are identical.*

The split-spread dataset was blended with blending factor 2, using a random time delay code. The delays were in the range of 0.1 s to 0.5 s and can be seen in figure 6.2 for each blended shot. Shot 1 was blended with with shot 77, shot 2 with shot 78 and so on. Pseudo-deblended common shot/receiver gathers can be seen in figure 6.3. These correspond to the unblended gathers of figure 6.1.

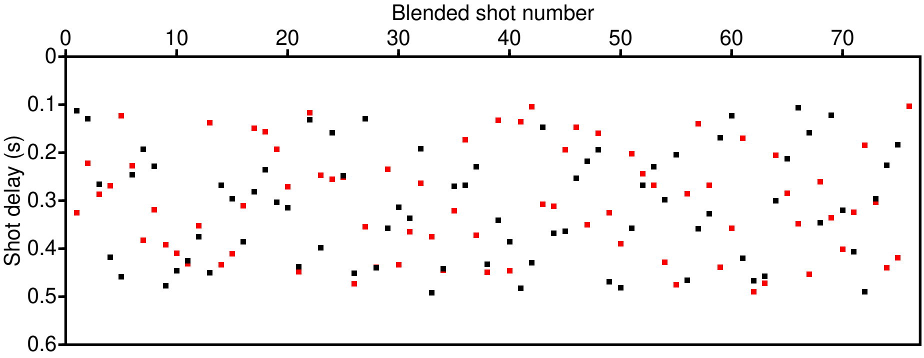


Figure 6.2: *The blending code for the 2D line example. Each column contains the delay times for a particular blended shot record. Red and black dots represent the first and second shot, respectively.*

Before the data can be deblended, a set of focal operators must be defined. Using NMO analysis, the 13 velocities and zero-offset traveltimes shown in table 6.1 were picked. Each velocity–zero-offset traveltime pair defines a flat reflector depth via the relationship $z = ct_0/2$, where z is the depth, c the velocity and t_0 the traveltime. Focal operators are constructed using the velocity–depth pairs and the expressions found in table 3.1. The focal grid for each operator is also a 2D line at each depth level below the acquisition, with a spacing of 25 m. For each point of the focal grid, the negative and positive local subsurface offset is arranged such that the total aperture is aligned with the acquisition and is 3750 m in length.

The deblending results after 500 SPGL1 iterations can be seen in figure 6.4. A first break mask was used in the inversion, a tool which can be very effective in suppressing blending noise above the early events. Additionally, events away from zero-offset/zero-time were penalized using a weighted version of the ℓ_1-norm, as described in chapter 3.5. Its effect is clearly visible in figures 6.6a-c as ellipsoid-shaped clusters of high-magnitude samples.

From the difference panels shown in figure 6.5, it can be seen that the blending noise leakage is relatively minimal for this example and the seismic events present in the data have been properly reconstructed. Most of the blending noise leakage seems to be related to aliasing. The signal-to-blending noise ratio for the result is 16.79dB.

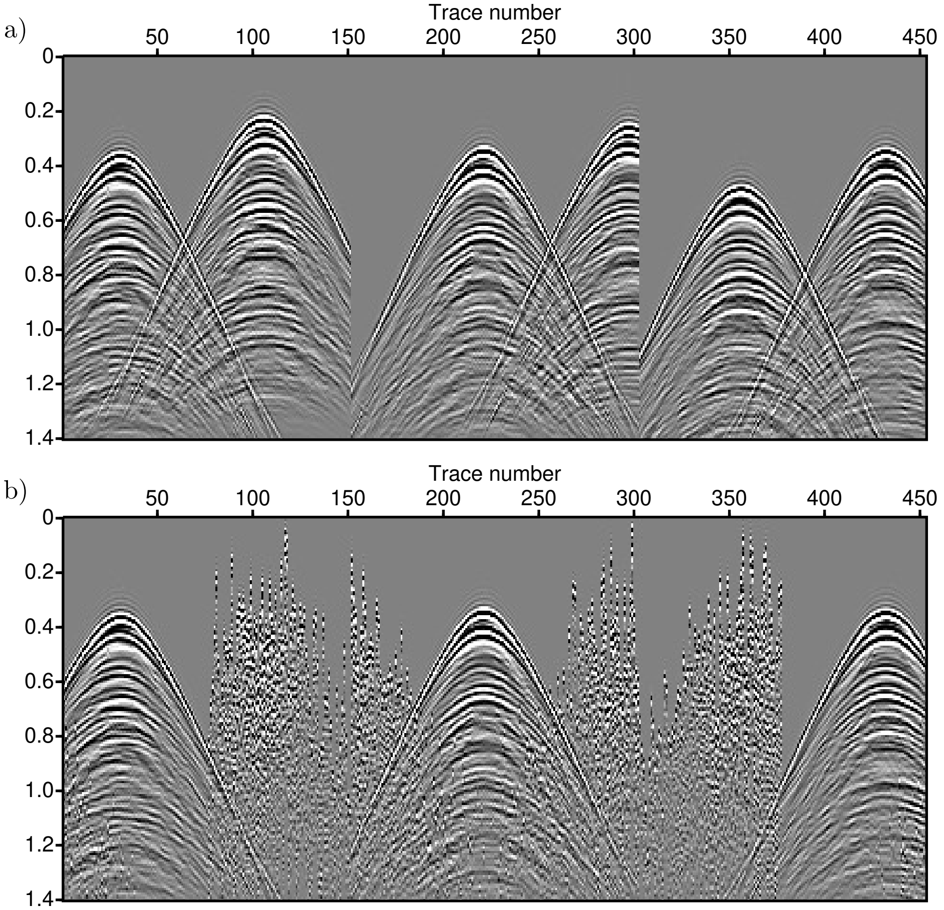


Figure 6.3: *2D line example, pseudo-deblended a) common shot gathers, b) common receiver gathers.*

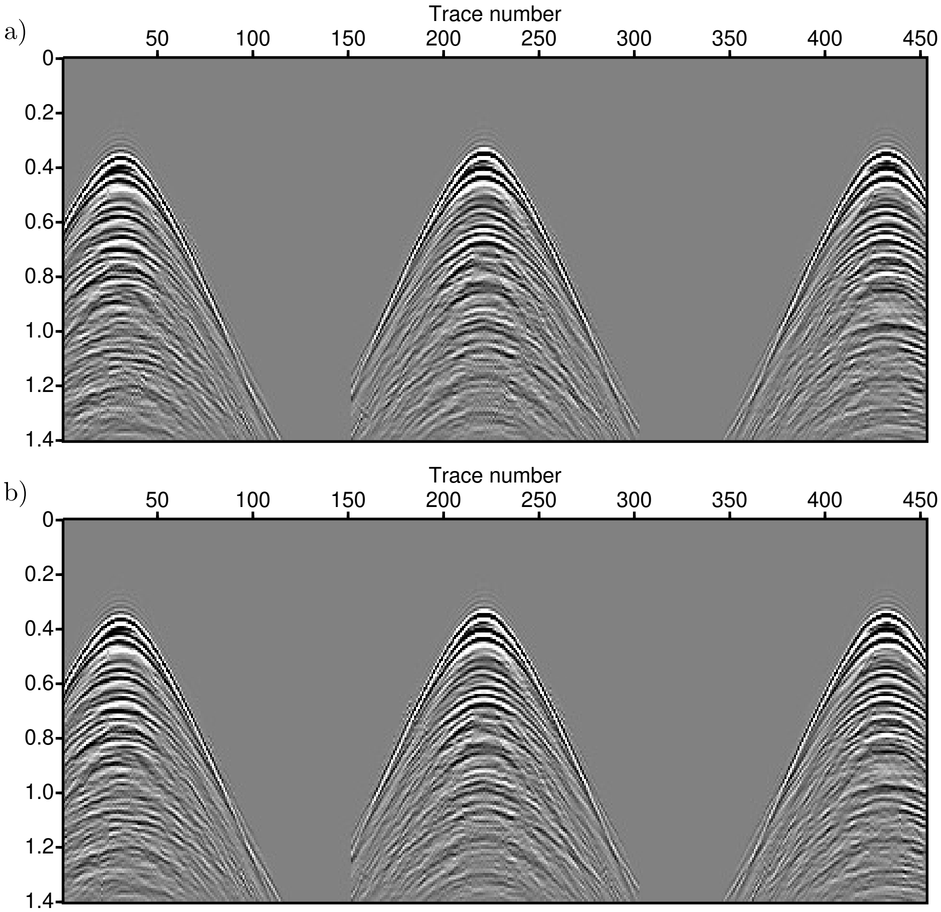


Figure 6.4: *2D line example, deblended a) common shot gathers, b) common receiver gathers.*

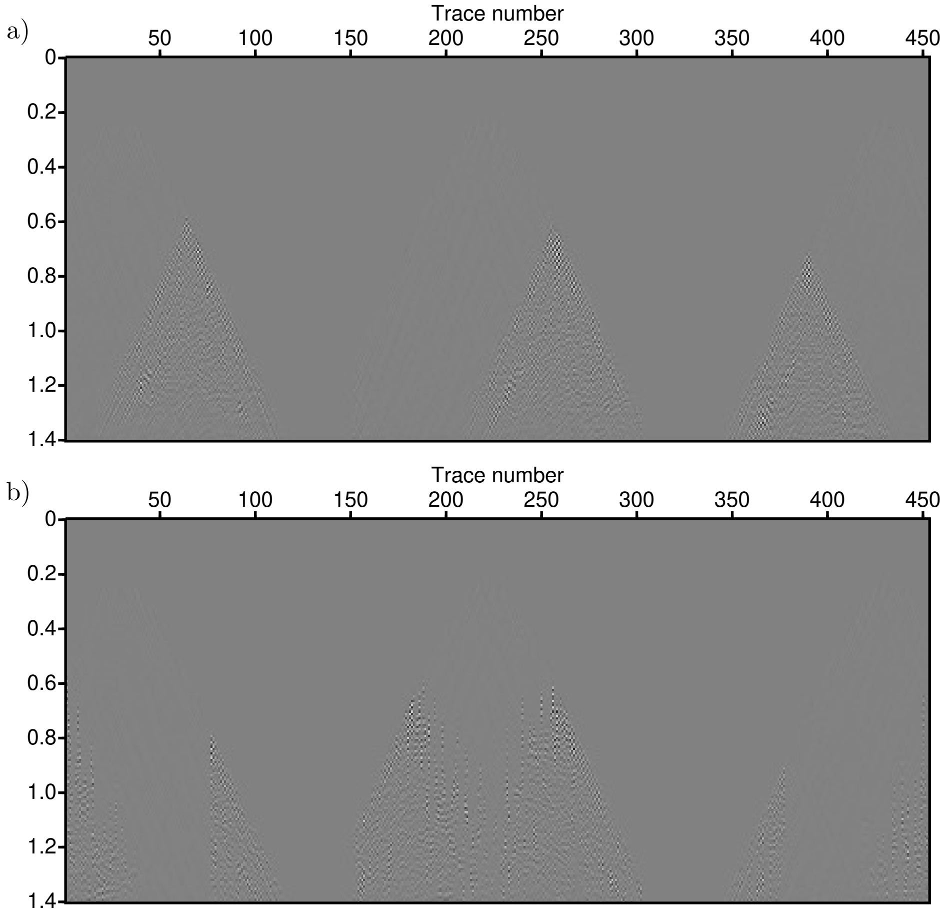


Figure 6.5: *2D line example, difference between unblended and deblended data, a) common shot gathers, b) common receiver gathers.*

Table 6.1: *Velocities and corresponding zero-offset traveltimes/depths for the focal operators for the 2D line example.*

Operator #	Velocity (m s^{-1})	Traveltime (s)	Depth (m)
1	1550	0.3440	267
2	1600	0.4182	335
3	1680	0.5630	473
4	1700	0.6264	532
5	1720	0.6500	559
6	1720	0.7228	622
7	1740	0.7951	692
8	1810	0.9370	848
9	1840	1.0440	960
10	1860	1.1400	1060
11	1860	1.2130	1128
12	1880	1.2910	1214
13	1880	1.4460	1359

Figure 6.6 depicts a part of three focal subdomains. The focusing effect is clearly visible in this example, with most energy clustering around $t = 0$ s for each of the subdomains.

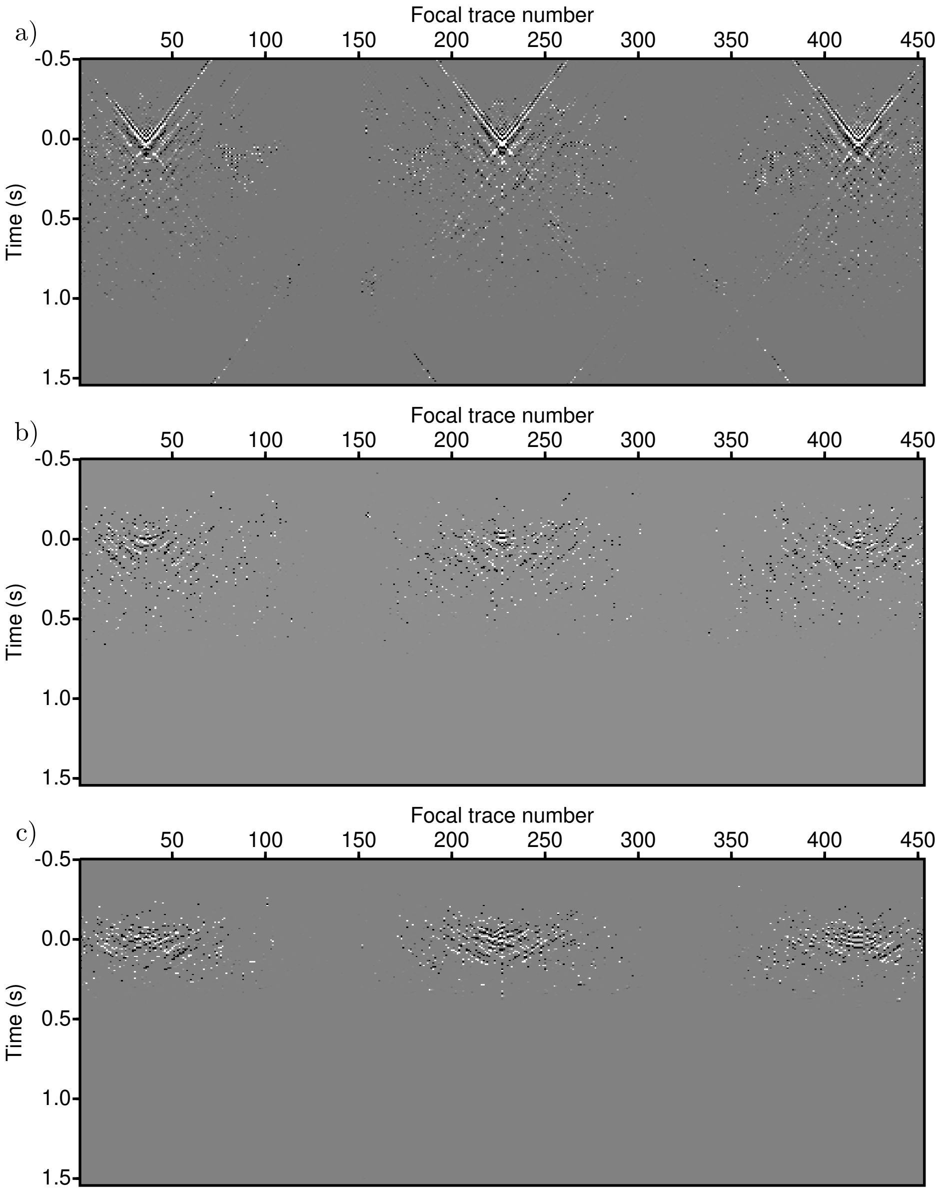


Figure 6.6: *2D line example, part of the a) first, b) fifth, c) ninth focal subdomain.*

6.2 Smart subsets: Streamer data

The next example explores the idea of processing subsets of the blended acquisition separately, a concept discussed in more detail in chapter 4.2. This makes parallelization of the deblending procedure easier, and reduces its computational effort. This happens because only a subset of the data is involved when the focusing/defocusing operators are applied. Here we take a look at an example on streamer data. Each each streamer can be thought of as an independent dataset and in this case the blending/pseudo-deblending can be applied entirely independently as well.

The dataset used here was provided by CGG. The streamer configuration can be seen in figure 6.7. I extracted a single streamer with 300 receivers, for 96 shots. The receiver spacing in the dataset was 12.5 m, for a total of 3750 m maximum offset/streamer length. The shot spacing was 18.75 m.

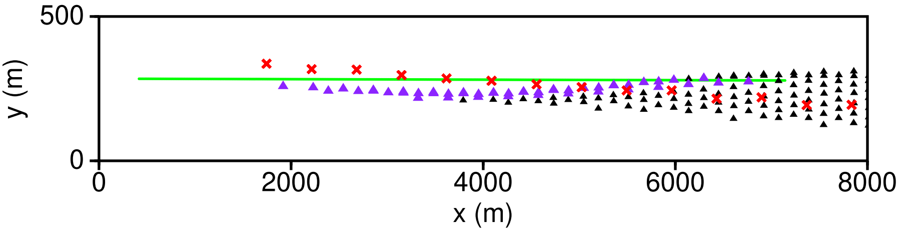


Figure 6.7: *Part of the acquisition used in the streamer deblending experiment, in rotated coordinates. Red crosses: shot locations; Black triangles: receivers on streamers; Purple triangles: receivers on the streamer participating in the experiment; Green dots: focal grid.*

The data was numerically blended using delays ranging from 2.1 s to 2.5 s. The exact delays can be seen in figure 6.8, each column representing a blended shot gather and each color the delay of an individual shot. The blending code was arranged such that there is overlap of at most three shots. The chosen range of delays means that in principle the early events of one shot gather will tend to overlap with the late events of another. Three shot gathers from the unblended data can be seen in figure 6.9a. The same three shots after pseudo-deblending are shown in figure 6.9b.

The double focal transform is used as well for this deblending example. The velocities and depths used for defining the operators can be seen in table 6.2. The focal grid is a single straight line, approximately parallel to the streamer, as marked in figure 6.7. 539 grid points with a spacing of 12.5 m were used. For each grid point, local subsurface offset points up to 2 km were calculated by the focal operator, using a spacing of 12.5 m. The local offset points also lie on the same line as the focal grid.

As mentioned earlier, the delay range of the blending code leads to early events with

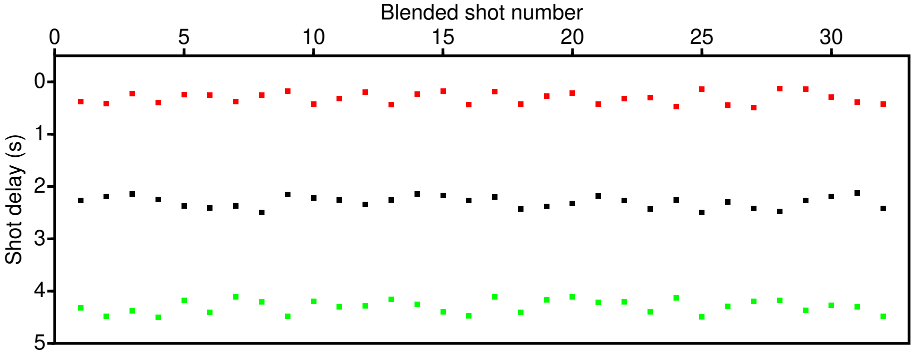


Figure 6.8: *The blending code for the streamer example. Each column contains the delay times for a particular blended shot record. Red, black and green dots represent the first, second and third shot respectively.*

Table 6.2: *Velocities and corresponding zero-offset traveltimes/depths for the focal operators used in the streamer example.*

Operator #	Velocity (m s^{-1})	Traveltime (s)	Depth (m)
1	1511	1.438	1086
2	1652	1.953	1613
3	1869	2.580	2411
4	1946	2.927	2848
5	1570	2.898	2275
6	2180	3.490	3804

relatively large amplitude being superimposed on later, weaker events. In this situation separation becomes more challenging. The deblended result and its difference from the unblended data, after 300 SPGL1 iterations, is shown in figures 6.10a and 6.10b. The separation was not perfect, as the imprint of the sea-bottom reflection and nearby events can still be seen after deblending. A positive observation, however, is that late events masked by strong blending noise have become visible again after separation. The signal-to-blending noise ratio of the result for this example is 21dB.

Figures 6.11a-c show a part of the focal subdomains. The lack of negative offsets in the streamer data leads to focal subdomains that look more asymmetrical than cases where negative offsets are present in the blended data, compare e.g. with figure 6.6. This is particularly visible for the energetic early events with amplitude variations along the offset dimension. Deeper events are more concentrated in subsurface offset and the highest magnitudes cluster near the zero time mark as expected.

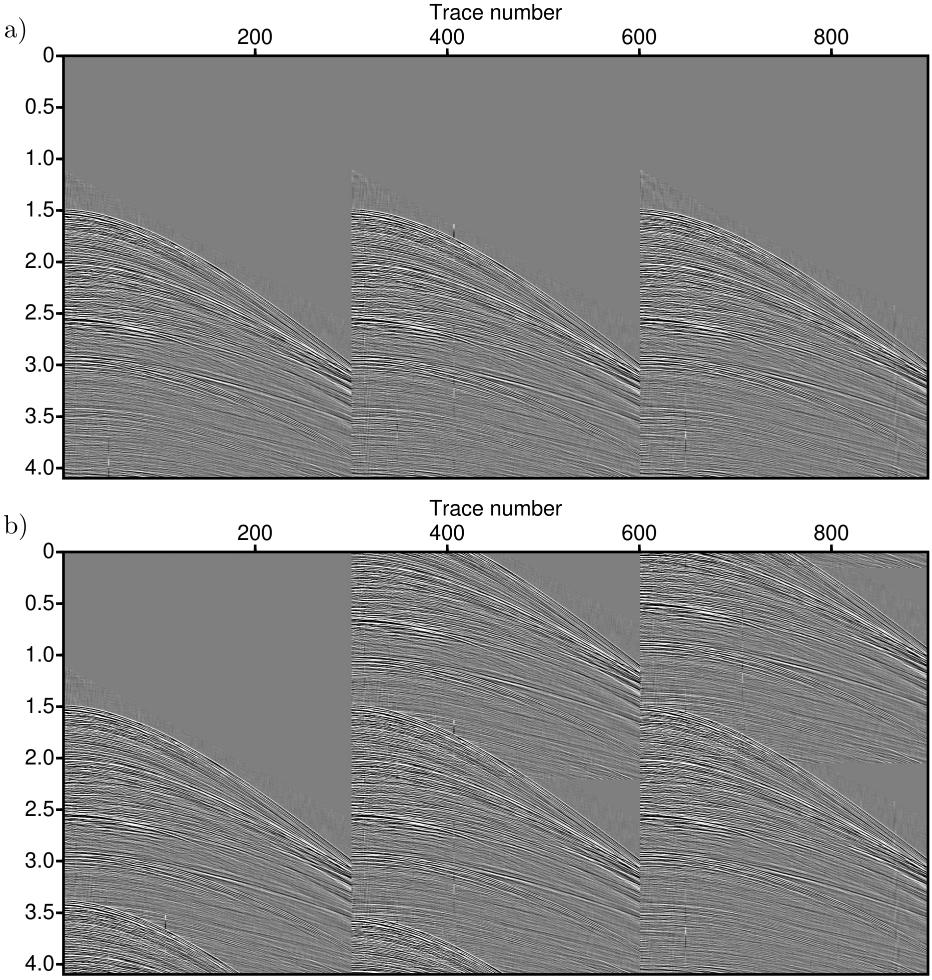


Figure 6.9: *Streamer example, a) unblended data, b) pseudo-deblended data.*

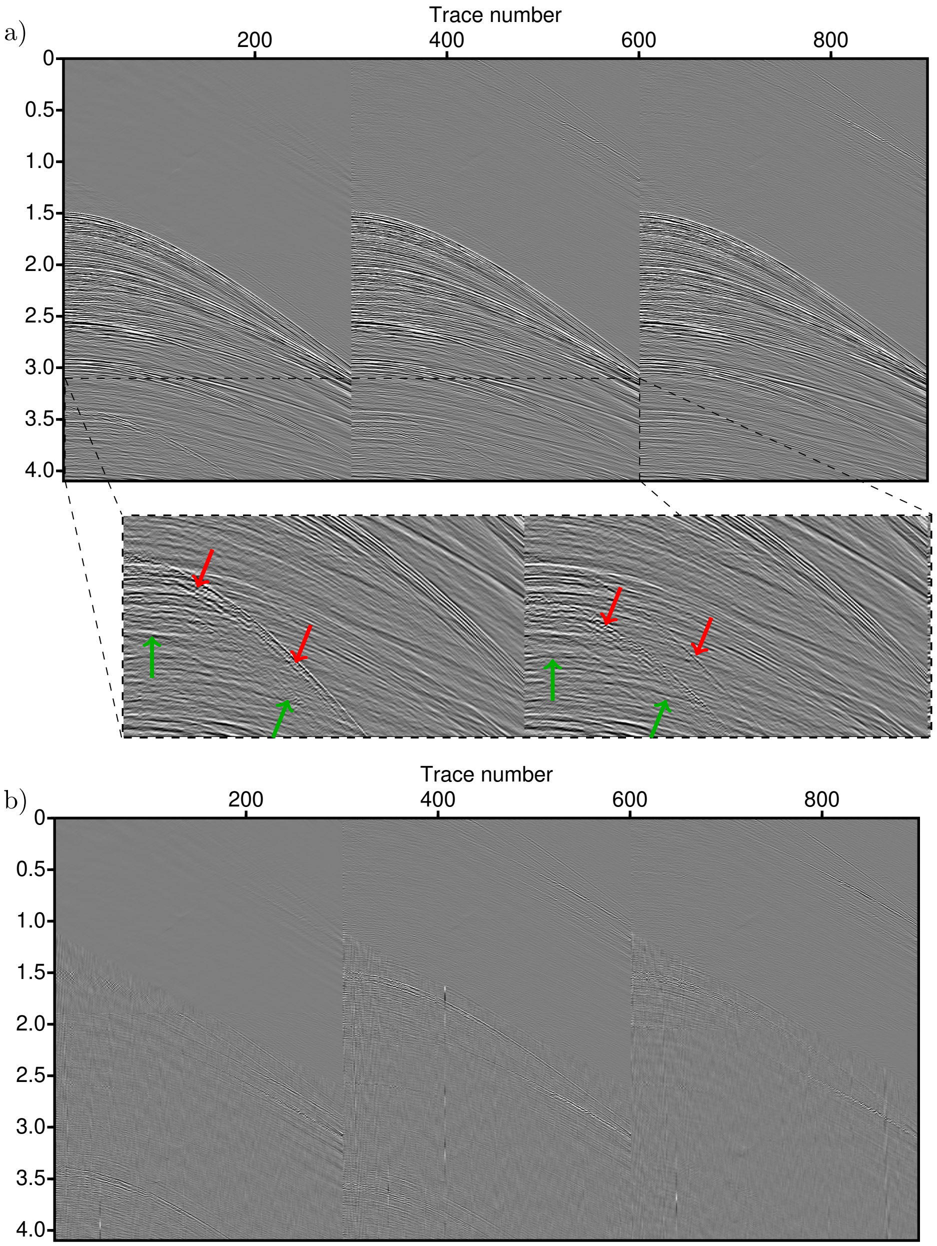


Figure 6.10: *Streamer example, a) deblended data, b) difference between unblended and deblended data. Red arrows indicate blending noise leakage, green arrows indicate recovered events previously masked by blending noise.*

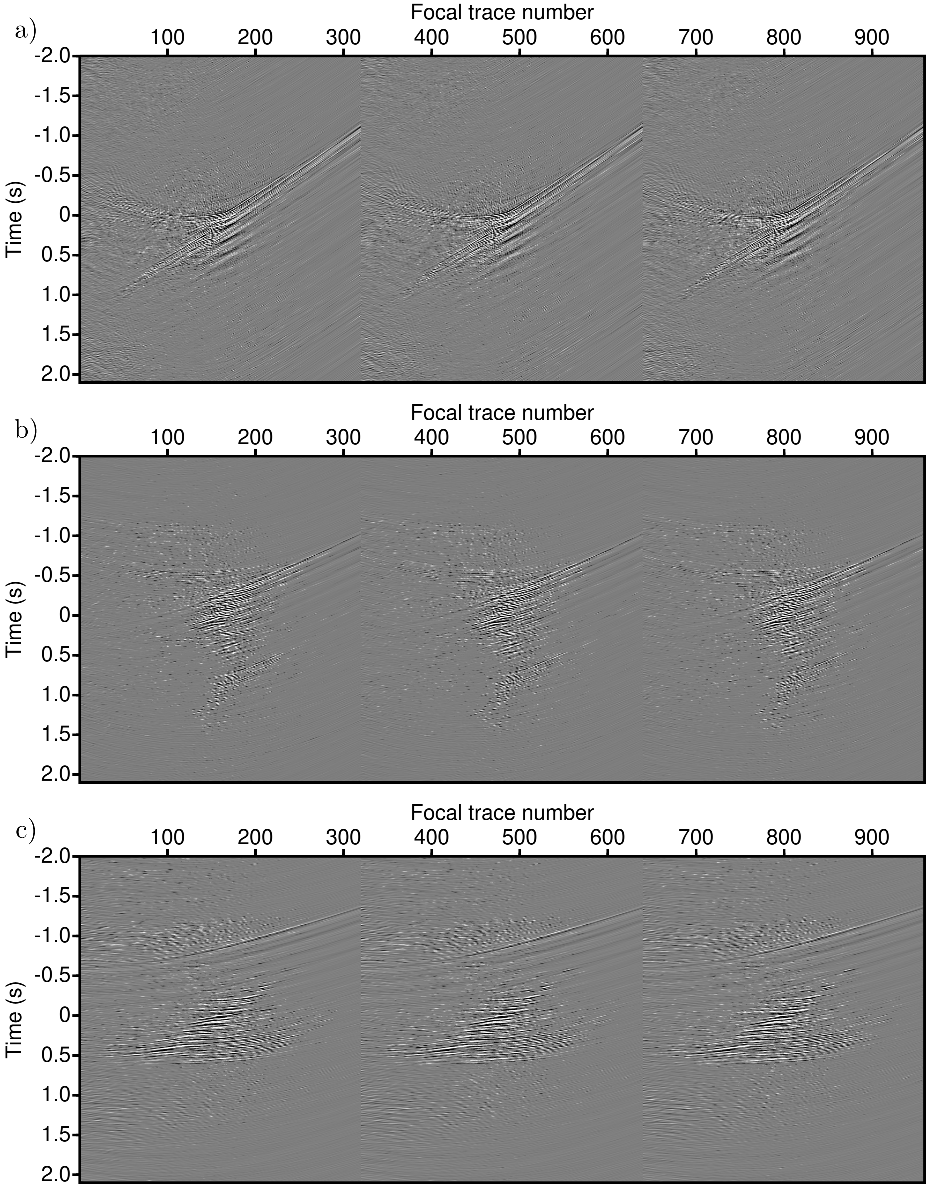


Figure 6.11: *Streamer example, part of the a) first, b) third, c) sixth focal subdomain.*

6.3 Smart subsets: OBN data

The next example also explores the idea of dividing the dataset into subsets before applying focal deblending, this time for OBN data [Kontakis and Verschuur, 2017a], acquired by GEOMAR and provided by TEEC. Figure 6.12 depicts the so-called race-track shooting acquisition layout. From the total acquisition, I extracted two shot lines of 700 locations each, the corresponding traces all associated with a particular receiver node. The selected receiver node and shot lines are marked with green and red color, respectively, in figure 6.12. For the OBN case, each common receiver gather naturally forms an independent subset. The survey was carried out with a dense shot sampling of 6.25 m, on average. Note that I subsampled traces by a factor of 3 for plotting purposes in the figures mentioned below.

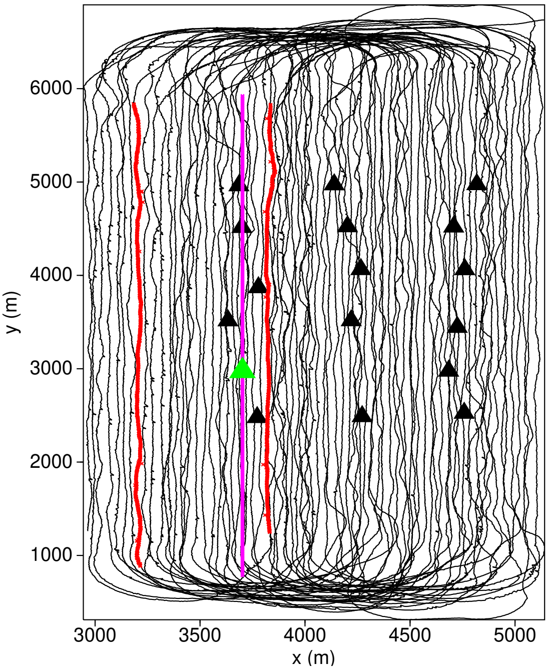


Figure 6.12: *Acquisition layout for the ocean bottom node example. The black triangles and black lines indicate the nodes and source trajectories, respectively. The extracted shot locations appear in red and the associated receiver node in green. The used focal grid is shown in magenta.*

The acquisition was done conventionally, without blending. To create a blended dataset, I assume that there are two source vessels, each of which takes care of its own part of the source locations. Each shot was assigned a random time delay code with delays ranging from 0.1 s to 0.4 s. The delays for each blended shot gather can be seen in figure 6.13. The original, unblended data as well as the data after pseudo-deblending are shown in figures 6.14a and 6.14b. The time sampling interval was 4 ms and 3 s of data are kept.

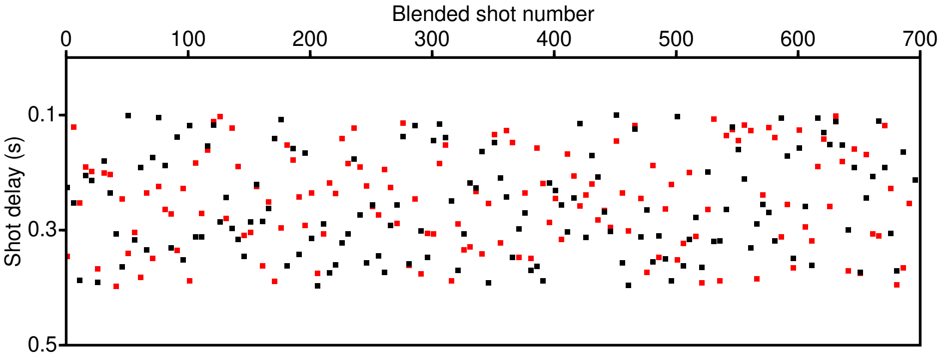


Figure 6.13: *The blending code for the OBN example. Each column contains the delay times for a particular blended shot record. Red and black dots represent the first and second shot respectively.*

A line passing through the chosen receiver node and parallel to the extracted part of the survey acts as a focal grid. In total 735 grid points with a spacing of 7 m are used, for a total of 5138 m. The points are highlighted with a magenta color in figure 6.12. The grid points in this line define also the subsurface offset sampling. Since this dataset involves only one receiver, the single-sided focal transform is used for deblending. Five focal subdomains, defined by the velocities listed in table 6.3, were used for focusing/defocusing. Note that the given depth values in table 6.3 are only approximations based on the NMO velocities.

After 600 SPGL1 iterations, the deblended result seen in figure 6.15a is obtained. Traces from three out of five focal subdomains are plotted in figures 6.16a-c. The difference from the unblended data in figure 6.15b shows some leakage around the sea bottom reflection event, however most of the blending noise has been effectively suppressed. The remaining blending noise has a laterally inconsistent character, which makes it unlikely to significantly affect migration. The signal-to-blending noise ratio for this example is 19.8dB.

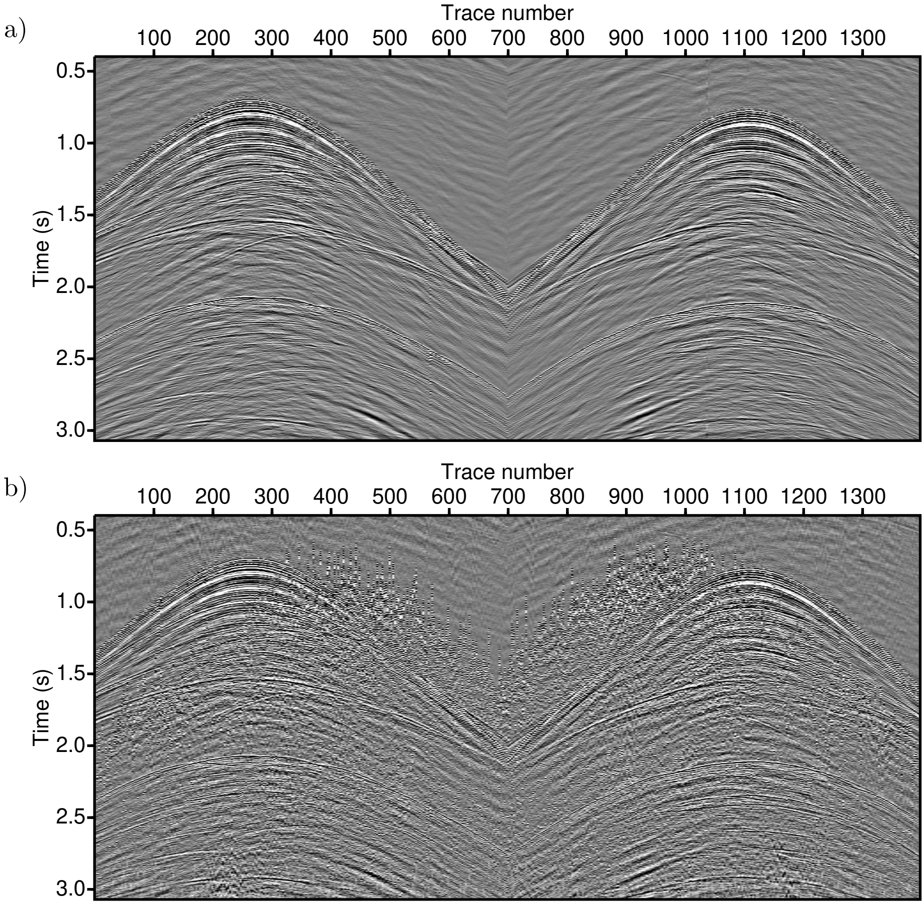


Figure 6.14: *OBN example, a) unblended data, b) pseudo-deblended data. The visible spatial aliasing effect is only visual, as only one out three traces is plotted.*

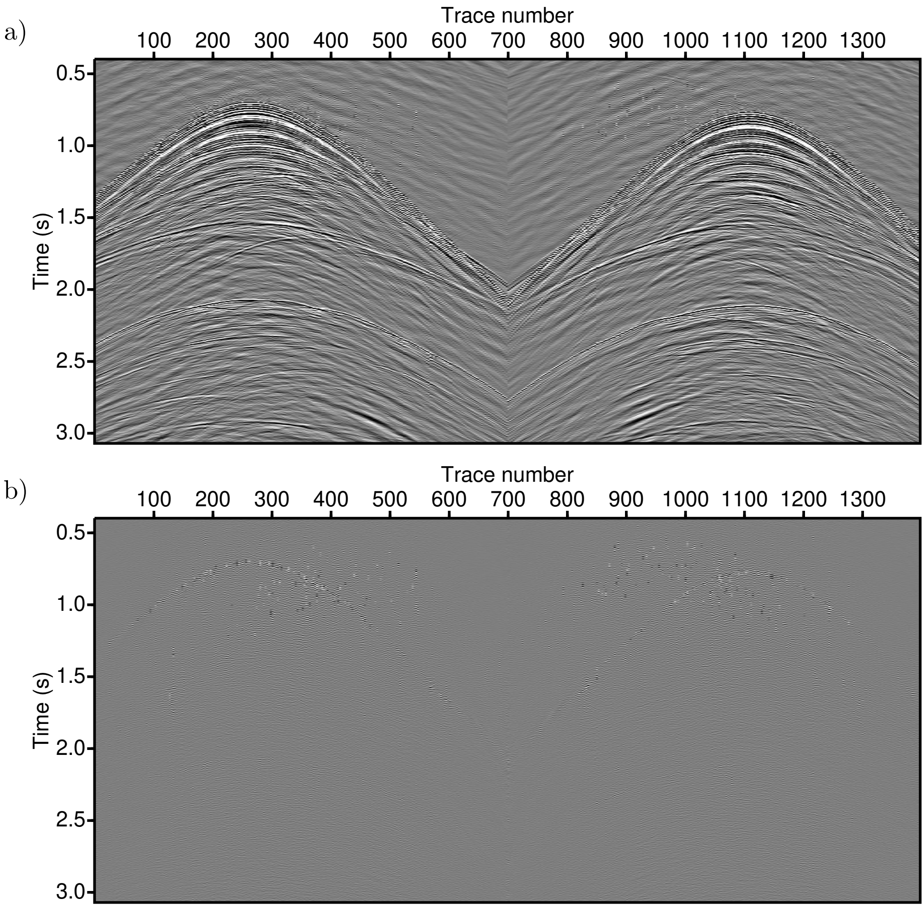


Figure 6.15: *OBN example, a) deblended data, b) difference between unblended and deblended data, amplified by a factor of 5.*

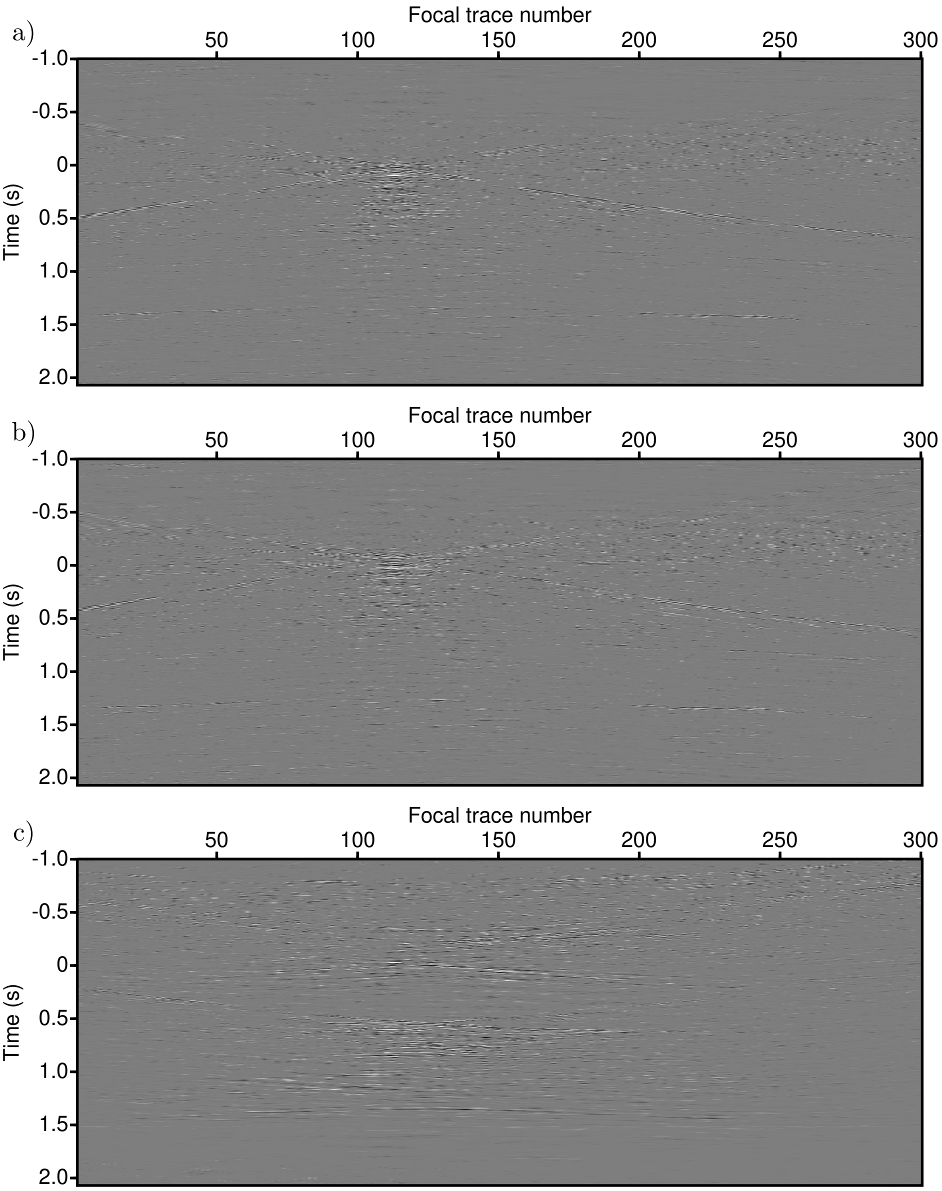


Figure 6.16: *OBN example, part of the a) first, b) second, c) fifth focal subdomain.*

Table 6.3: *Velocities and corresponding zero-offset traveltimes/depths for the focal operators used in the OBN example.*

Operator #	Velocity (m s^{-1})	Traveltime (s)	Depth (m)
1	1482	0.700	519
2	1449	0.740	536
3	1426	0.770	549
4	1593	1.000	797
5	1800	1.540	1386

6.4 Hybrid transform

The next example shifts focus to the hybrid focal-curvelet transform [Kontakis and Verschuur, 2017b]. Part of a dataset provided by CGG is used to test the ability of the hybrid transform to deal with parts of the wavefield that cannot be explained by the focal transform. The dataset used in this example was already preprocessed: the direct wave was removed, reciprocity was used to create negative offset data and the near-offset gap was filled using parabolic Radon interpolation [Kabir and Verschuur, 1995]. A common shot gather was then extracted, consisting of 801 traces with a receiver spacing of 12.5 m and a time sampling interval of 4 ms. The receivers were arranged as a 2D line with 5 km of positive and 5 km of negative offset, for a total of 10 km of aperture. The shot and receiver coordinates were then exchanged to create a pseudo-common receiver gather. Two identical copies of the common receiver gather were blended using random time delays in the range 0.3 s to 0.8 s (figure 6.17). The unblended and pseudo-deblended gathers can be seen in figure 6.18. Note than in these as well as in following figures, traces have been subsampled by a factor of 3 for plotting purposes.

The blended common-receiver gather was deblended using the focal, the hybrid and the curvelet transform. The main idea behind this experiment is to test whether the hybrid transform can handle problems that might arise from an unsatisfactory choice of velocities for the focal transform part. In all the experiments 5 focal operators were used, defined by the velocities and depths listed in table 6.4. The focal grid was also a 2D line that was formed by translating the receiver coordinates to each of the depth levels listed in table 6.4. As such, the spacing between focal grid points is 12.5 m. For each of the focal grid points, the rest of the points serve to define the local offsets at which the focal transform is evaluated. The single-sided focal transform was used in this case. Regarding the curvelet transform parameters, the 2D version of the transform was used, with 8 scales and a maximum of 32 wedge angles. A weighted sum-norm was used for the inversion, the weights designed such that events away from zero time/zero offset are given less importance.

The deblended common receiver gathers after 100 SPGL1 iterations can be seen in

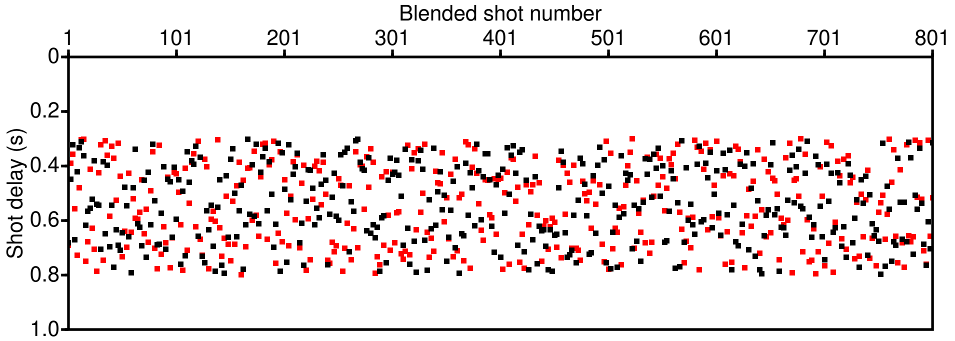


Figure 6.17: *The blending code for the hybrid transform example. Each column contains the delay times for a particular blended shot record. Red and black dots represent the first and second shot, respectively.*

Table 6.4: *Velocities and corresponding zero-offset traveltimes/depths for the hybrid transform example.*

Operator #	Velocity (m s^{-1})	Traveltime (s)	Depth (m)
1	1600	1.258	1006
2	1700	1.558	1324
3	1900	1.758	1670
4	1950	1.990	1940
5	2100	2.359	2476

figure 6.19 for each of the three deblending approaches. The differences between the unblended and deblended gathers can be seen in figure 6.20. Part of the first and third focal subdomains are depicted in figure 6.21a and 6.21b. The effect of the weighted ℓ_1-norm is evident, with most energy in the short local subsurface offsets and around $t = 0$ s. Figures 6.21c and 6.21d show the magnitude and phase content for each of the 32 wedges in the fourth curvelet scale. These are relatively sparse, as a result of the weights used for the focal part. What remains for the curvelet transform to explain is the far offset part of events and local amplitude variations not captured by the focal transform.

Comparing the focal transform result with the hybrid transform result (figures 6.20a and 6.20b), it can be seen that the low velocity event has been deblended more accurately when using the hybrid transform. This is expected, because the first focal operator has a rather high velocity. The curvelet part of the hybrid transform has dealt with this problem, yielding an improved result. An exception is the refracted wave event at the far positive offset, that has not been reconstructed by either of

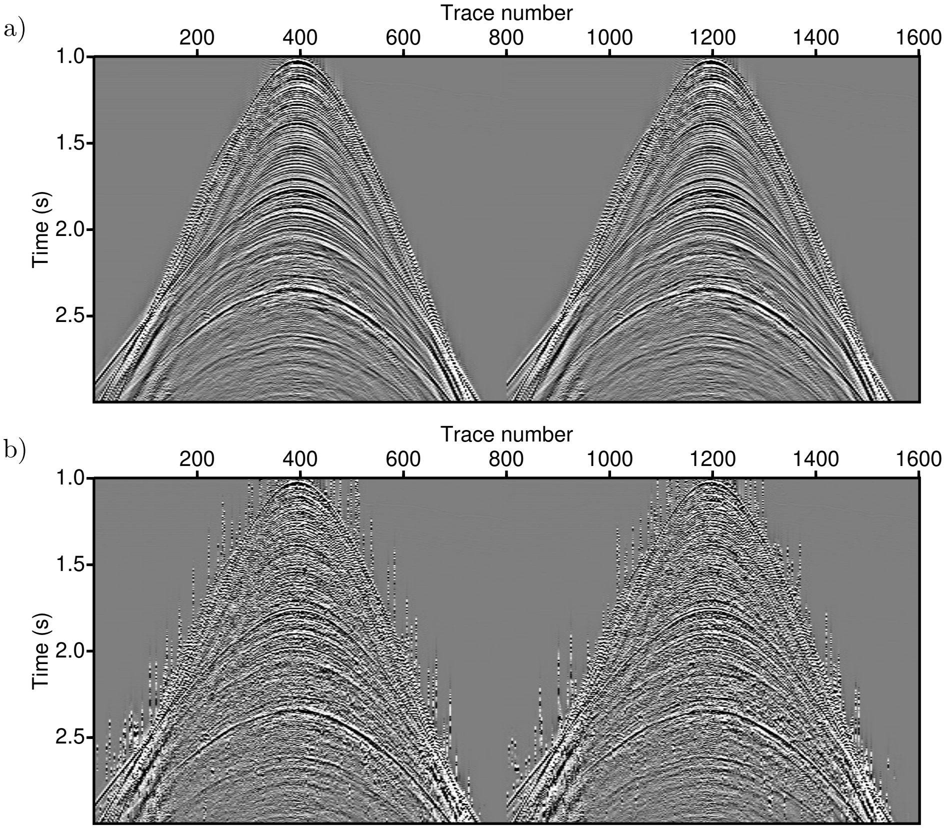


Figure 6.18: *Hybrid transform example, a) unblended common receiver gathers; b) pseudo-deblended gather. Only one out of three traces are plotted, creating a visual aliasing effect.*

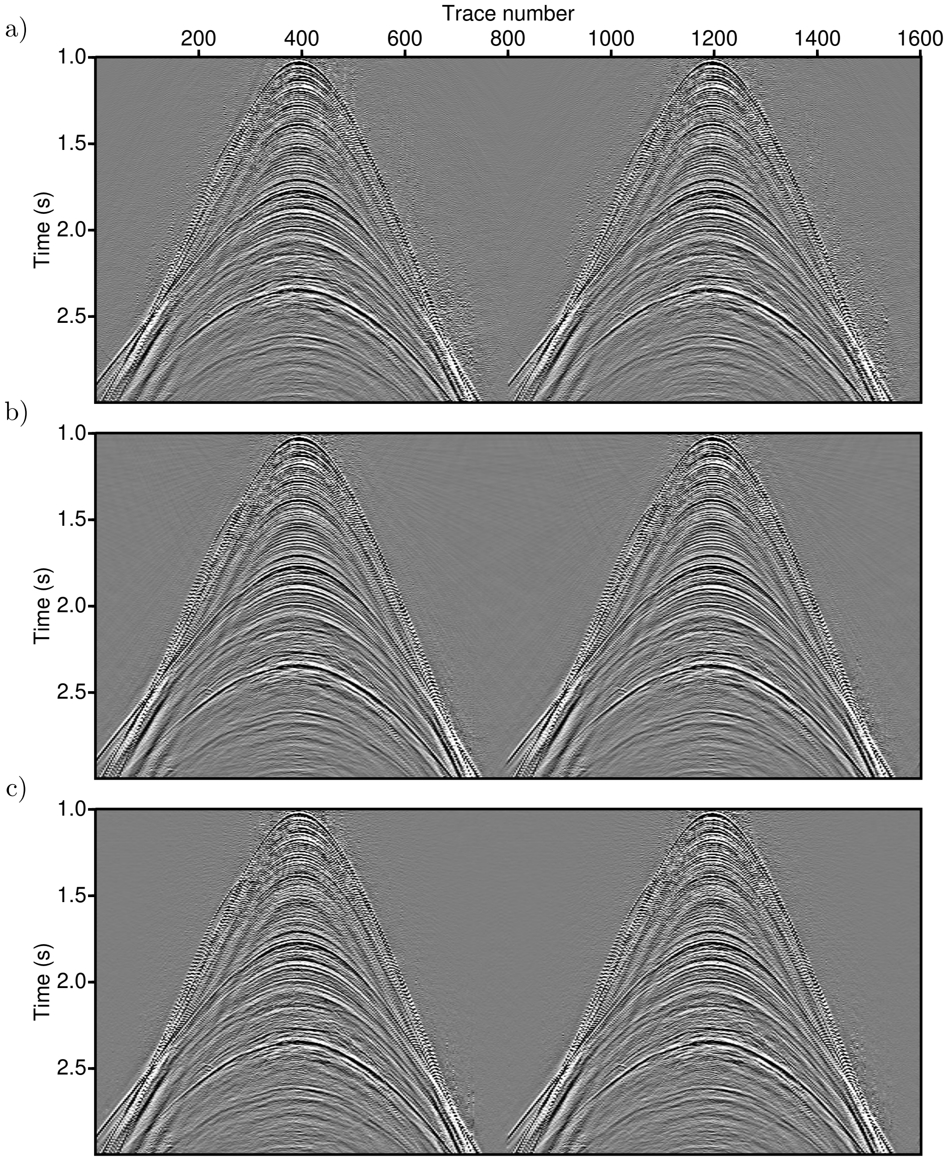


Figure 6.19: *Deblended gathers for the hybrid transform example, using a) only the focal transform, b) the hybrid transform, c) only the curvelet transform.*

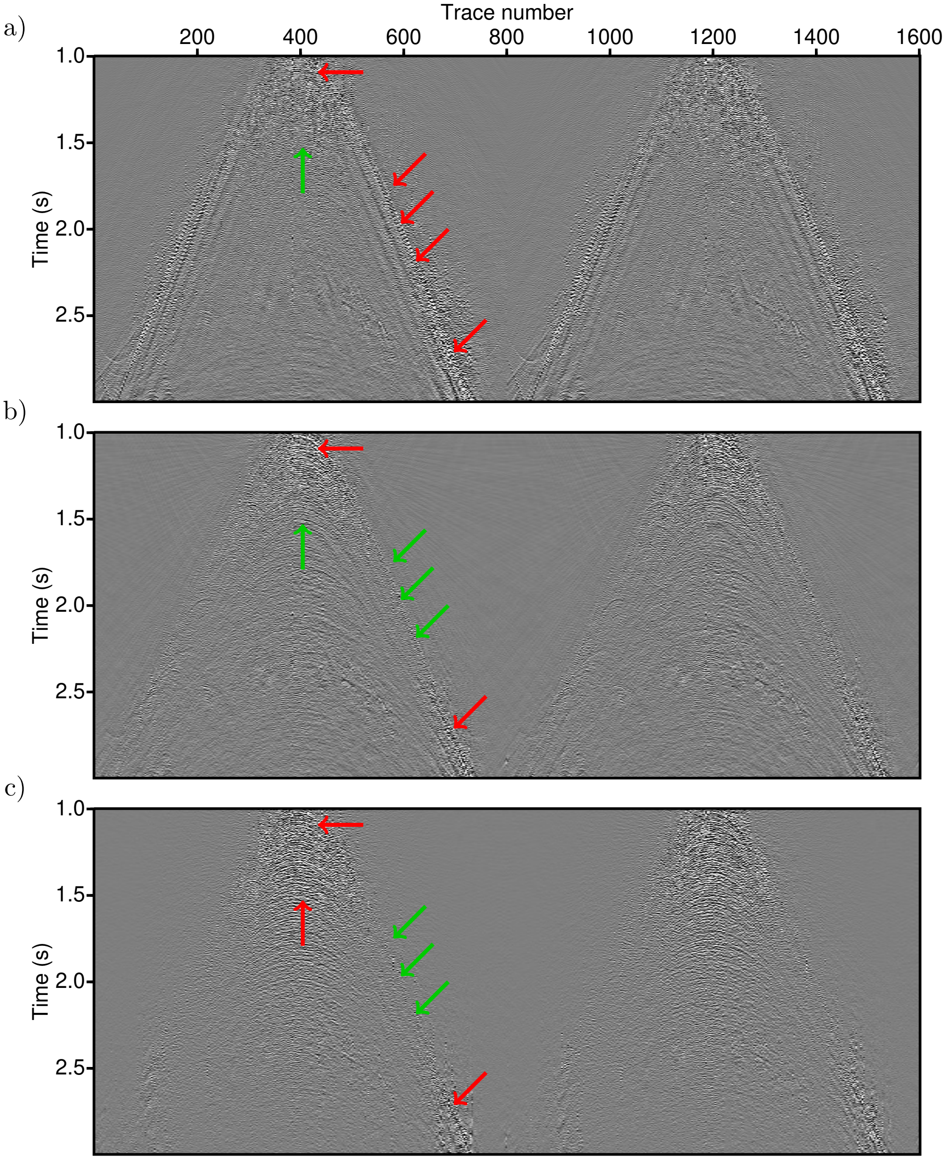


Figure 6.20: *Difference between the unblended and deblended gathers for the hybrid transform example, using a) only the focal transform, b) the hybrid transform, c) only the curvelet transform.*

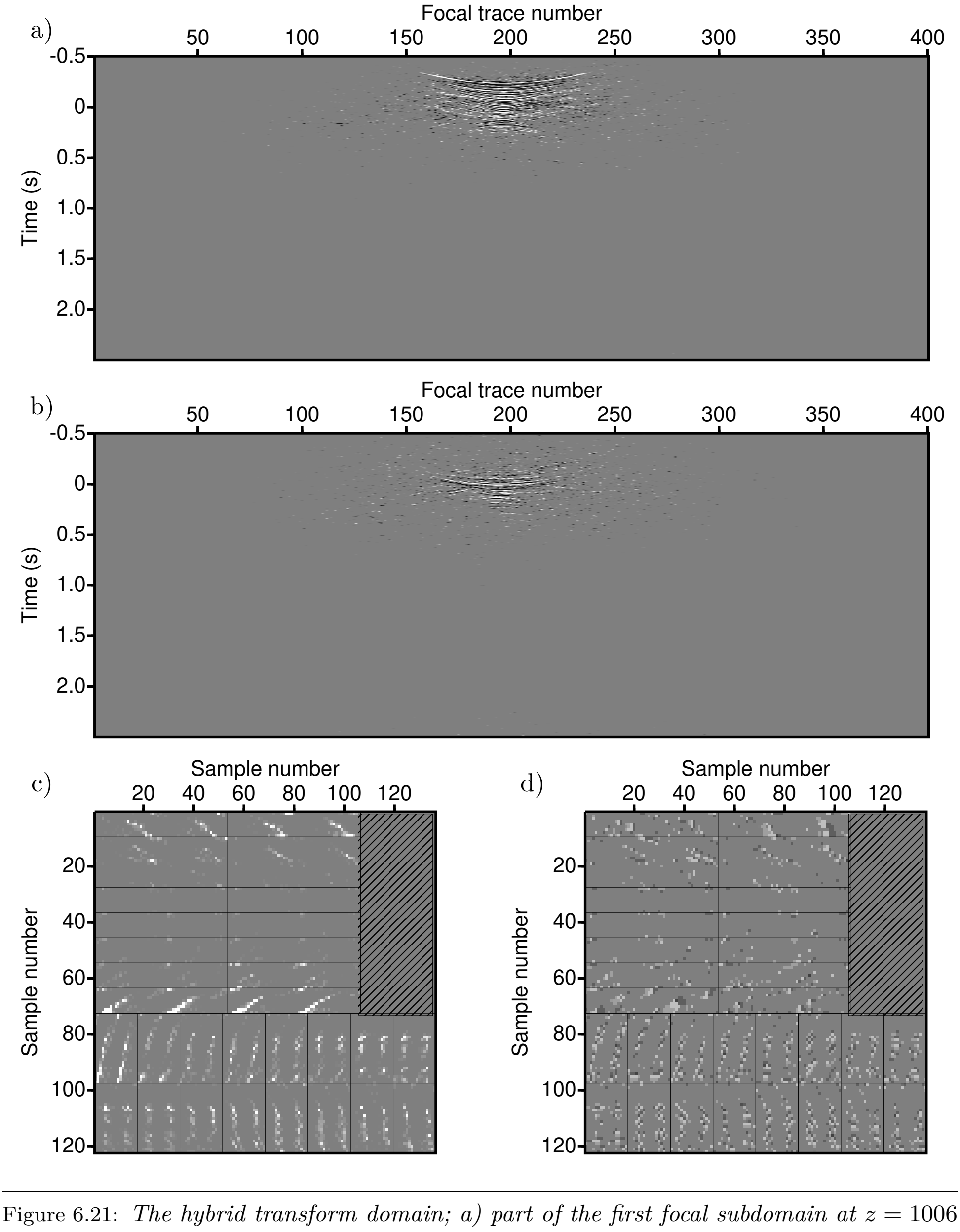


Figure 6.21: *The hybrid transform domain; a) part of the first focal subdomain at* $z = 1006$ *m, b) part of the third focal subdomain at* $z = 1670$ *m, c) magnitude of the wedge segments in the fourth curvelet scale, d) phase of the wedge segments in the fourth curvelet scale.*

the transforms. This could be improved by increasing the number of iterations and perhaps modifying the curvelet parameters.

The comparison between the hybrid and curvelet-only cases (figures 6.20b and 6.20c) demonstrates something seen also in the synthetic example of chapter 5: although the curvelet part performed well in explaining the low velocity event at the far offsets, the hybrid transform generally did better in the near offset region. Now it is the focal part of the hybrid transform that comes to aid, with its better performance at explaining the curved near-offset part of the seismic events. The signal-to-blending noise ratio was 8.98dB, 13.3dB and 11.6dB for the focal-only, hybrid and curvelet-only transform cases, respectively.

6.5 Field-blended

The next deblending example is conducted on a subset of the Carlsen 3D dataset, provided by TGS. Unlike the previous examples, in which numerical blending was used, here the data were blended in the field, as a result of the time interval between subsequent shots. The marine streamer geometry used in the acquisition is depicted in figure 6.22. Three airguns were fired sequentially, the first and second activated 33 times and the third 32 times, yielding a total of 98 common shot gathers in the subset. The average inline spacing between shots was 37.5 m. The streamer inline spacing for the hydrophones was 12.5 m. Focal deblending was used to deblend up to an offset of 2 km (145 traces per shot), where events exhibit most curvature and the focal transform is most effective at deblending.

Each shot was fired on average every 5.5 s, with random time dithering in the range of approximately -0.7 s to +0.7 s. Shot records with a length of 11 s were extracted from the continuous recording. The extraction took place starting from the time instant each source was activated, which then became time zero for the extracted record. This procedure is equivalent to pseudo-deblending as defined in chapter 2.3, followed by truncation of the pseudo-deblended records at the desired length. The firing times of each shot participating in a pseudo-deblended shot record can be seen in figure 6.23. As the firing times shown are relative, the shot record considered as signal always has a relative time delay of 0 s. The shot records appearing as blending noise have positive delays. In principle there is also blending noise from previous shots. The assumption made here is that blending noise from previous shots is negligible.

Three different shot gathers before preprocessing are shown in figure 6.24, each from a different airgun. The deblending problem is challenging as strong events overlap weak events of the previous shot. In the first half second, the direct wave and a strong, low-frequency event dominate in amplitude. The latter is probably related to the airgun bubble effect. These events were treated prior to deblending through preprocessing, which consisted of using a window between the time break

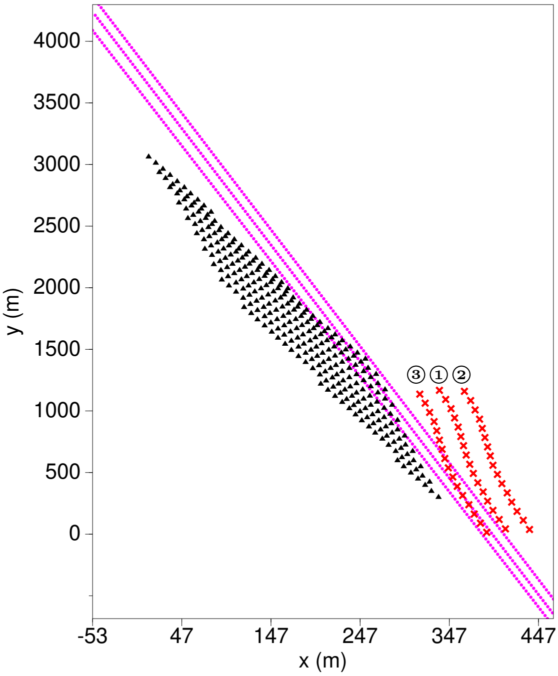


Figure 6.22: *Part of the field-blended acquisition. Red crosses: shot locations; black triangles: receivers on extracted streamer; magenta dots: focal grid points for each airgun subset. The numbers indicate the activation pattern used by the three airguns. Note that the x-axis is exaggerated here for plotting purposes.*

of the direct wave and the sea bottom reflection. Gabor filtering was used within this window to suppress the energetic low-frequency noise. Such events will not be effectively focused by the focal transform and will tend to dominate the inversion due to their high amplitude. As such, it is better to remove them before hand to prevent additional blending-noise leakage.

To keep the numerical effort and amount of memory required at an affordable level, I performed low-pass filtering up to 45 Hz, followed by subsampling the dataset from 4 ms to a 12 ms time sampling interval. The result of this preprocessing can be seen in figure 6.25. Comparing with figure 6.24, the presence of the direct wave and

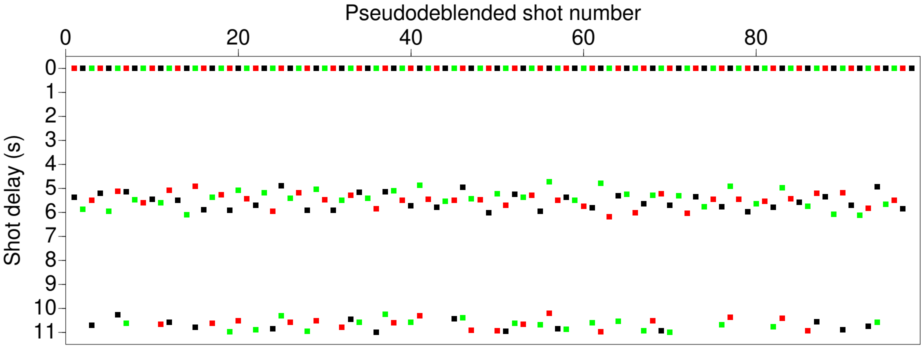


Figure 6.23: *The blending code for the field-blended data example. Each column contains the delay times of each shot present in the time window of a particular pseudo-deblended gather. The red, black and green dots correspond to shots by airgun 1, 2, and 3, respectively, as marked in figure 6.22.*

low-frequency noise is greatly reduced within the first half second. Since I applied this processing per common shot gather with a known blending code, the events removed by the processing are also removed from the blended gather copies, such that the end result remains consistent with the blending equation.

The preprocessed data were given as input to focal deblending. I divided the dataset in three subsets, each subset containing all shot gathers associated with the same airgun. I then constructed a focal grid for each subset by sampling a linear segment fitted using the common midpoint coordinates pertaining to each subset. The focal grid points were spaced 5 m apart along the linear segment, the total length of which was approximately 5 km. At each grid point the locations of all other points on the same line within a 500 m radius were used to define local subsurface offsets. The focal grids used are depicted in figure 6.22 as magenta points. The subset arrangement used here is “pseudo-2D”, in the sense that it consists of a line of shots, a line of receivers and a line grid for the focal transform, similar to what would happen for a 2D line survey. In reality the subset deviates from being truly 2D due to the presence of some feathering and the fact that the shot and the receiver lines have some distance between them. Placing focal points on a line rather than a grid reduces computational cost, but can come at the expense of less focusing for events that originate from the same depth plane but away from focal grid line.

Each subset had an independent set of focal subdomains defined for it. The associated focal operators used the same velocities/depths for all three subsets, but different focal grids for each subset. The list of velocities and depths for the 8 focal operators are given in table 6.5. The moveout for a focal operator may approximately match that of a multiple rather than a primary event. This in principle is not an issue, as long as the focal operator helps to reconstruct all events to be part

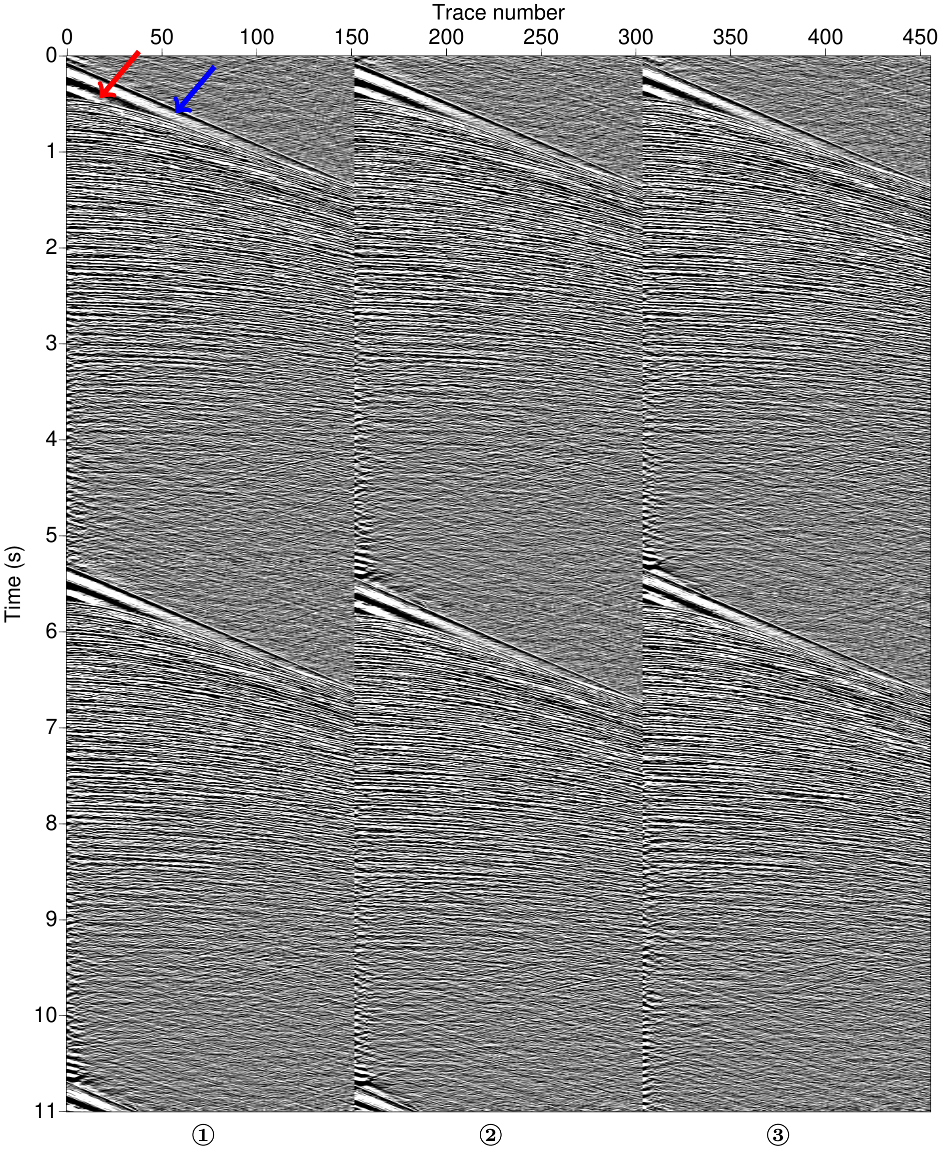


Figure 6.24: *Pseudo-deblended common shot gathers from the Carlsen 3D dataset, before preprocessing. The number below each gather indicates the airgun number it is associated with. The direct wave and a low frequency event, likely related to the airgun bubble, are indicated with a blue and red arrow, respectively.*

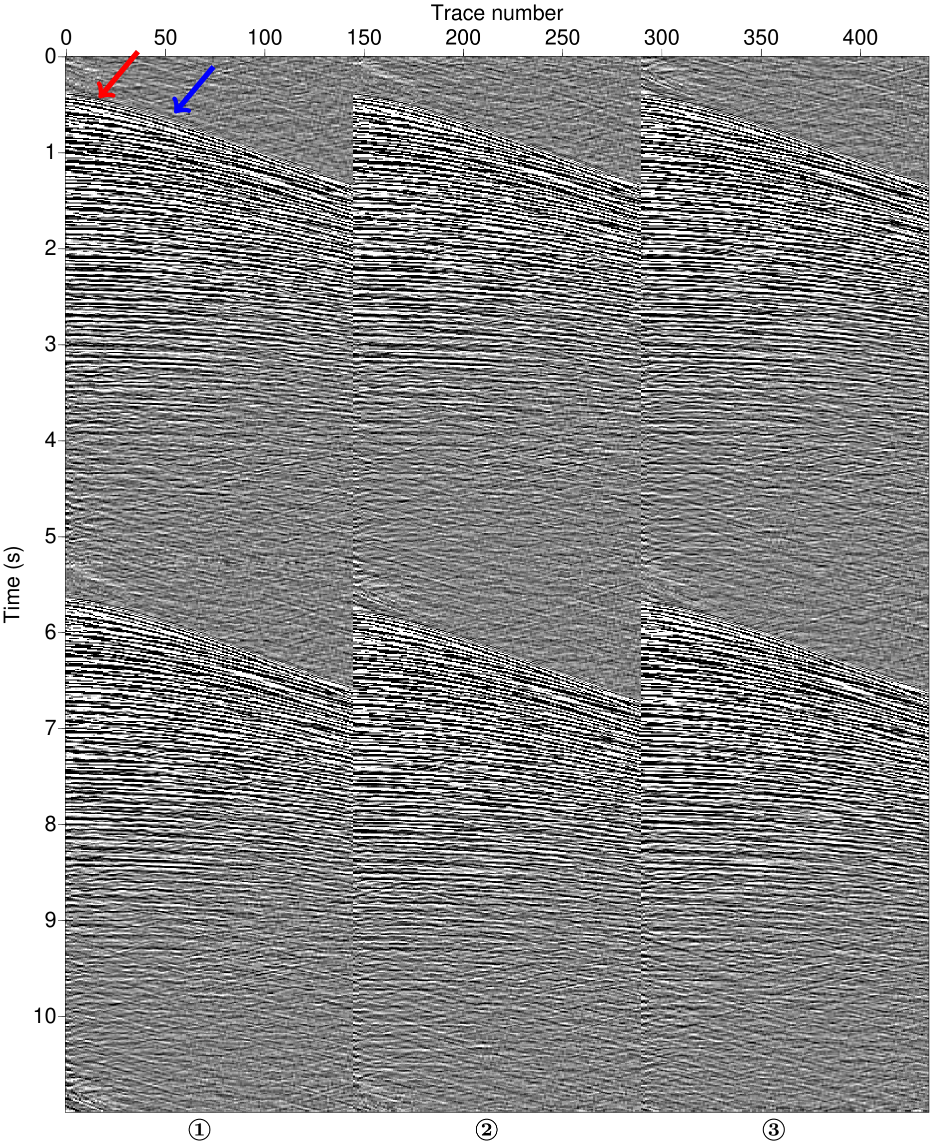


Figure 6.25: *Pseudo-deblended common shot gathers from the Carlsen 3D dataset, after preprocessing. The number below each gather indicates the airgun number it is associated with. The blue and red arrow indicate the locations of the removed events for the first gather.*

of the deblended data. I used the double focal transform for this inversion problem, and a total of 2000 SPGL1 iterations.

Operator #	Velocity (m s^{-1})	Traveltime (s)	Depth (m)
1	1480	0.46	339
2	1572	0.61	483
3	1677	0.76	634
4	1515	1.03	781
5	1499	1.49	1114
6	1589	1.75	1393
7	1577	6.00	4731
8	1670	8.00	6680

Table 6.5: *Velocities and corresponding zero-offset traveltimes/depths for the focal operators in the Carlsen 3D example.*

The majority of the events that are responsible for most of the blending noise are in the window from 0 s to 5 s. I took advantage of this fact to help minimize blending-noise leakage in two ways. The first was to use data-domain weighting and give higher weight to residuals of events occurring before 5 s. In this way they contribute more to the ℓ_2-norm of the residual, making it more likely that they will be explained before the penalized blending noise copies appearing after 5 s. The data-domain weights take the form of a sigmoid function having the value 1 before 5 s, transitioning to the value 0.1 after 5 s (figure 6.26a).

I also employed focal-domain penalty weights, in addition to data-domain weighting, to discourage inclusion of events occurring away from $t = 0$ s in the focal domain, where most blending noise is expected to be found. For this particular example, the weight increased linearly in the positive and negative time direction, starting from 1 at 0 s and ending at 10 at ±5 s. Note that the focal-domain weights are used to implement a weighted ℓ_1-norm, therefore unlike in the data-domain case, higher weights should be interpreted as higher penalty (figure 6.26b).

As a comparison the same deblending experiment was repeated using sparse curvelet deblending. The streamer receivers were binned according to their x-coordinate, forming bin gathers in which the blending noise was incoherent. The 2D curvelet transform was set up to use four scales and up to 16 wedge angles. Data-domain weights of the same form as the ones used in focal deblending were also applied in the curvelet deblending, except that they transitioned from the value 1 to the value 0.8 after the 5 s mark. A total of 600 SPGL1 iterations where used for curvelet deblending.

Three deblended gathers can be seen in figure 6.27 for focal deblending and figure 6.28 for curvelet deblending. These correspond to the gathers shown in figure 6.25. Despite the challenging blending problem, a lot of the blending noise has been

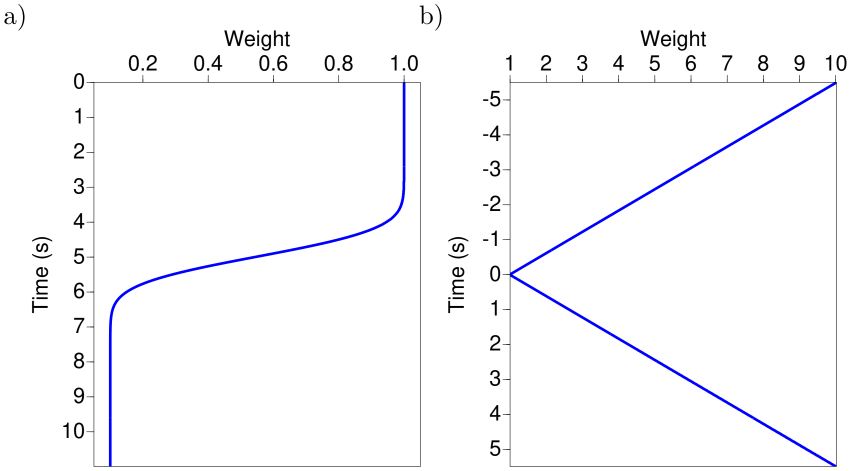


Figure 6.26: *a) Data domain weights; b) focal domain penalty weights. For data-domain weighting, lower values indicate higher data-fitting penalty. For focal-domain weighting higher weight values lead to larger penalties.*

removed by focal deblending, although there is some noticeable blending-noise leakage. This is especially evident in the near offset traces. The linear moveout events featured prominently from 5 s to 11 s in the deblended result are likely to be a combination of blending noise and transform-related artifacts. Still, a major reduction in blending noise is achieved and original reflection information is recovered that was not visible before. Comparing the deblended gathers from the two methods, it appears that the character of the leaked blending noise is different, with that from focal deblending being lower-frequency than that from curvelet deblending.

Figure 6.29 shows a common grid point gather extracted from four focal subdomains of the first airgun data subset. The presence of some amount of local subsurface offset is almost always necessary to be able to faithfully reconstruct seismic event amplitudes. This is evident from the fact that there is focal content present away from zero offset. Extracting the zero-offset trace for each grid point allows to form image-like gathers, as seen in figure 6.30. These can reveal subsurface structure and help redesign the focal grid, if needed. The last traces correspond to the grid points furthest from the shot locations. The lower amplitudes seen in this part of the figure hint at lower subsurface illumination. This is expected, as grid points get further away from source-receiver midpoints. For this particular example, these extra grid points were included to be able to reconstruct weak back-scattered events with opposite dip, present in the large offsets of the input data. From both figures it can be seen that the focal subdomains contribute to some extent to events away from their target depth.

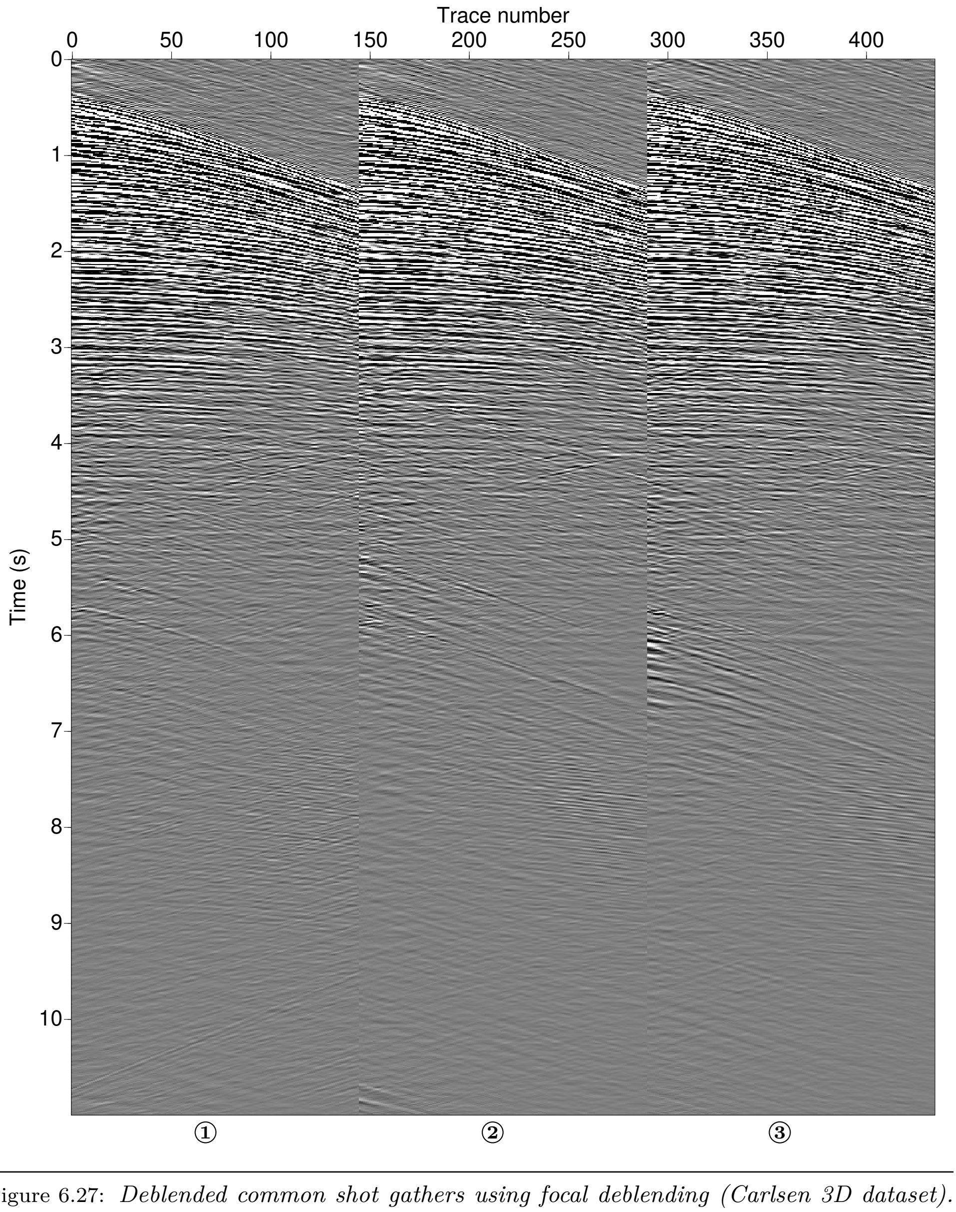


Figure 6.27: *Deblended common shot gathers using focal deblending (Carlsen 3D dataset). The number below each gather indicates the airgun number it is associated with.*

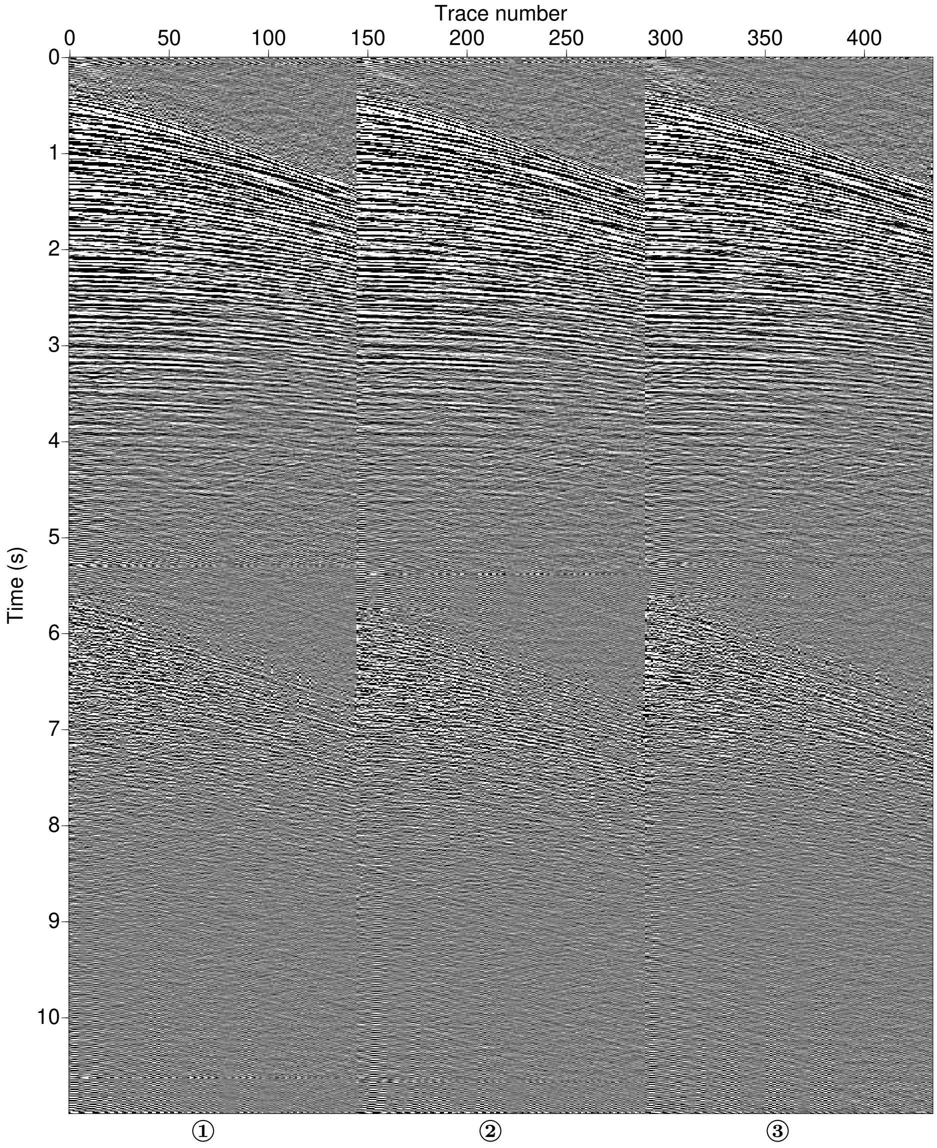


Figure 6.28: *Deblended common shot gathers using curvelet deblending (Carlsen 3D dataset). The number below each gather indicates the airgun number it is associated with.*

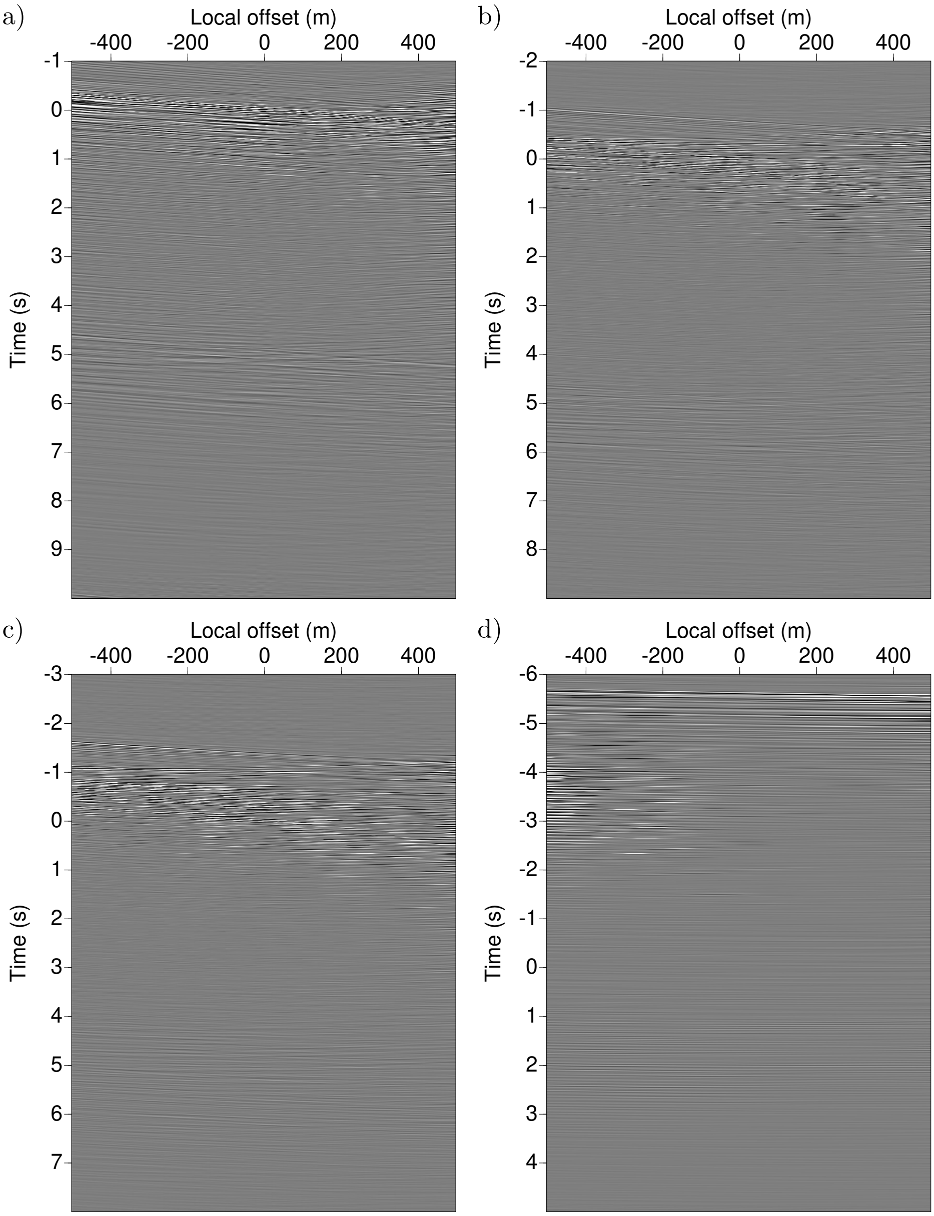


Figure 6.29: *Focal subdomain common grid point gather for the first airgun subset; a) first subdomain, b) third subdomain, c) fifth subdomain, d) seventh subdomain.*

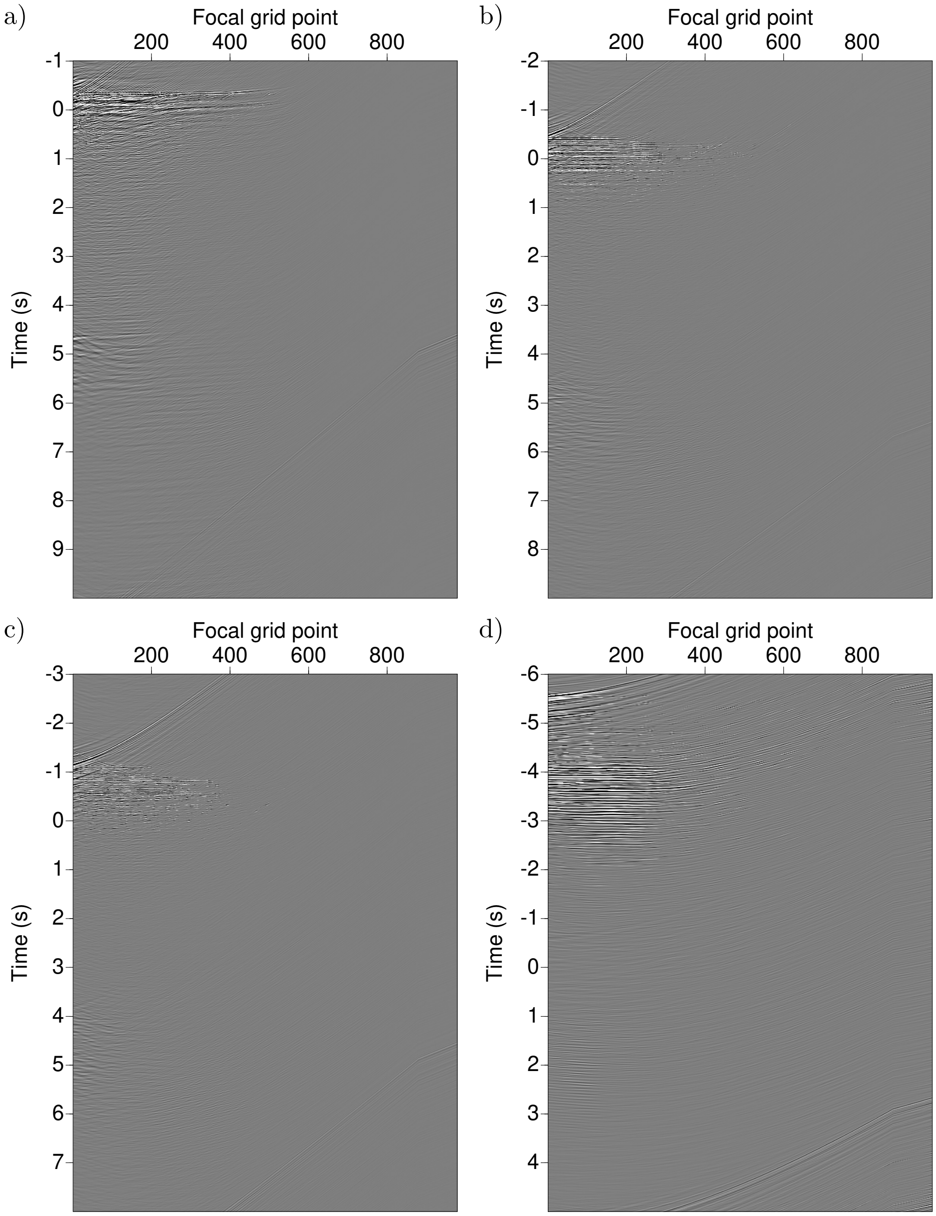


Figure 6.30: *Focal subdomain zero local offset gather for the first airgun subset; a) first subdomain, b) third subdomain, c) fifth subdomain, d) seventh subdomain. Note the different time axes for each plot.*

Events associated with deep reflectors can be very weak and thus hard to distinguish in prestack gathers. In order to assess the recovery of reflector-related information after deblending in the 5 s to 11 s time window, I applied stacking after NMO-correcting the data, for both deblending methods. The velocity model used for NMO correction can be seen in figure 6.31a, superposed on a semblance plot used for picking. After rotating coordinates such that they are aligned with the y-axis, I split the dataset into CMP bins with dimensions 100 m×12.5 m. I then summed the NMO-corrected traces within each bin to produce a stack gather. The fold for each stacked trace can be found in figure 6.31b.

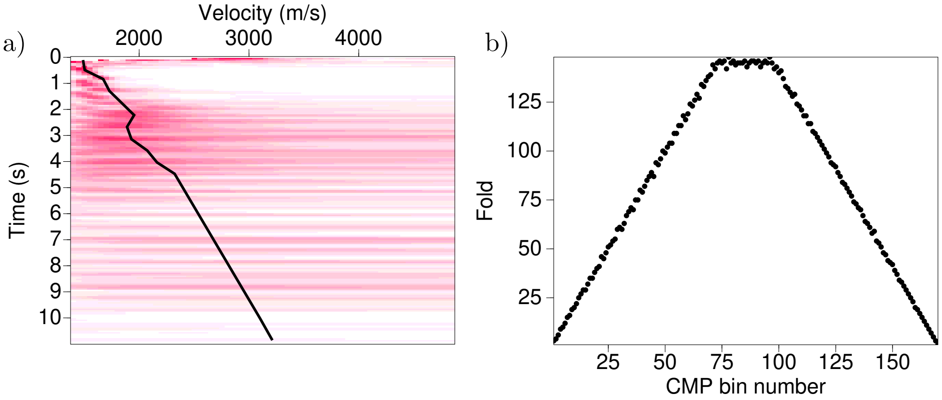


Figure 6.31: *a) Velocity profile used for NMO correction superimposed on a semblance plot; b) fold for each CMP bin.*

For comparison purposes, I repeated the same stacking procedure on the pseudo-deblended input data. The stacked input and stacked deblended output for focal and curvelet deblending are shown in figures 6.32a, 6.32b and 6.33b, respectively. I applied a linear time gain to amplify the deeper events. In both cases some blending-noise leakage is clearly visible even after deblending and is associated to the near-offset blending noise seen relatively consistently in most deblended gathers. Due to the low number of shots, the stacking power is not enough to reveal weak events in the areas marked by red arrows. In the mid offsets, however, certain reflectors that were completely masked by blending noise input become discernible. These are indicated with green arrows in figures 6.32b and 6.33b, showing the possibility to recover some subsurface information generating weak events even after the 5 s mark. This can be seen more clearly in figures 6.32c and 6.33c, where linear Radon and curvelet filtering has been applied to attenuate blending-noise leakage and transform artifacts. Note that the better separation achieved by focal deblending has produced reflectors that are more clearly visible in the middle part of the stack sections.

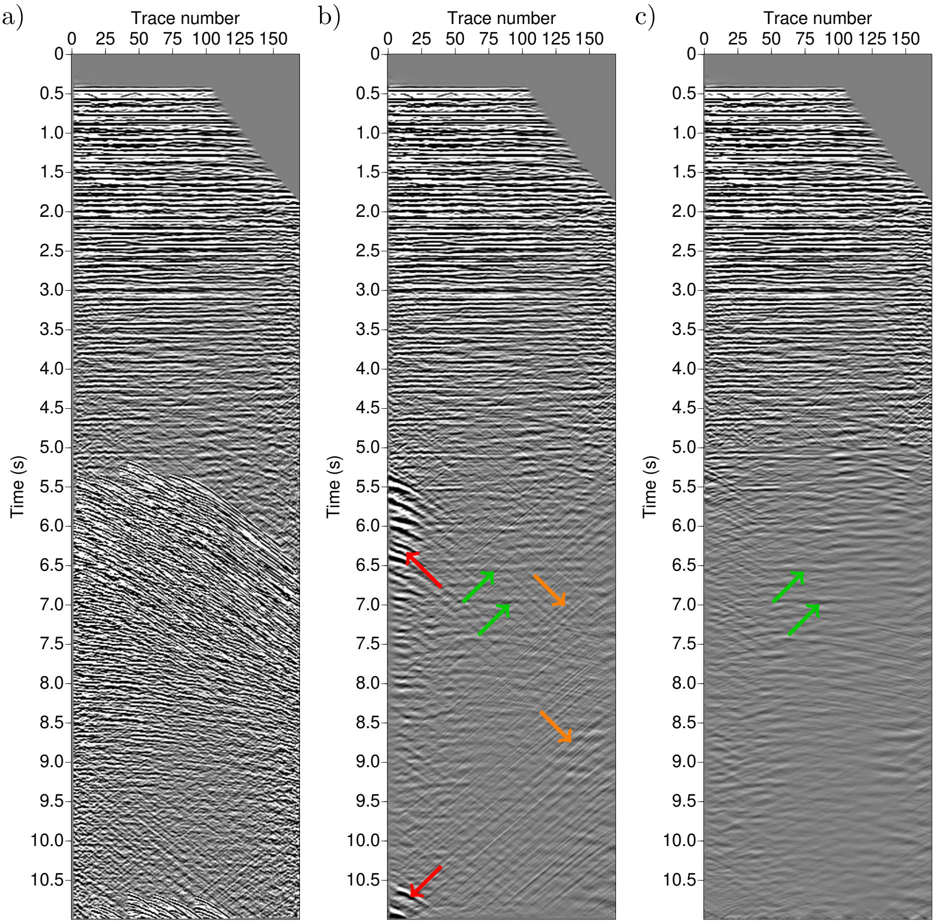


Figure 6.32: *Stacked data, focal deblending, a) before deblending, b) after deblending, c) after applying linear Radon and curvelet filtering on the deblended stack. Red arrows indicate areas of excessive blending noise leakage. Green arrows indicate recovered reflectors. Orange arrows point at what are likely transform artifacts.*

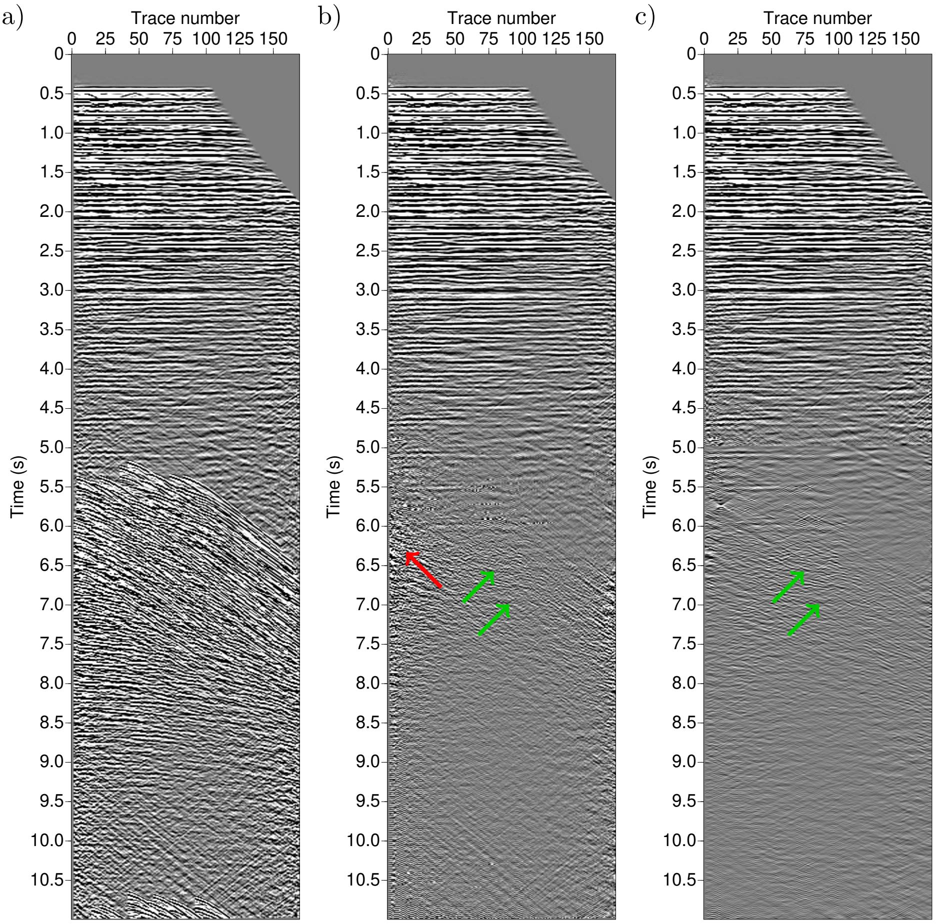


Figure 6.33: *Stacked data, curvelet deblending, a) before deblending, b) after deblending, c) after applying linear Radon and curvelet filtering on the deblended stack. Red arrows indicate areas of excessive blending noise leakage. Green arrows indicate recovered reflectors. Orange arrows point at what are likely transform artifacts.*

7

Conclusions, observations and discussion

7.1 Conclusions

The main objective of this thesis was to investigate an inversion-based method for deblending that employs the focal transform and sparsity-promoting optimization. In the course of developing and testing this approach to deblending, I made several observations that I discuss below. A number of these stem from the fact that the proposed method incorporates, as part of its machinery, concepts from the field of seismic migration. Repurposing migration-related tools for source separation opens up opportunities, but comes with a set of challenges of its own. A few of the other observations listed below are not tied to focal deblending in particular, but are pertinent to deblending algorithms in general.

■ 7.1.1 The properties of the blending code matter

Good autocorrelation/crosscorrelation characteristics mean that decoding, in the form of pseudo-deblending, can by itself suppress a considerable amount of blending noise. Because airguns are generally used in marine acquisition as sources, time delay codes are the most practical to implement. They are useful in that they introduce incoherence in the spatial coordinates, or purpose-crafted coherence [Robertsson et al., 2016a], which a deblending algorithm can take advantage of. However, they do not affect blending noise amplitude (i.e. no incoherence in the time direction), which might still be rather strong after pseudo-deblending.

■ 7.1.2 Implications of using a migration-like scheme for deblending

An important assumption that the focal deblending method makes is that the unblended signal can be represented as a sum of time/space shifted diffraction events that have propagated through a homogeneous acoustic medium. This assumption might not always hold, especially when the actual subsurface is complex and/or the velocities provided by the user are very inaccurate. Those events that cannot be represented in this manner will be absent from the deblended result, which may also lead to additional blending noise as shown in chapter 3.6.

For migration applications having a residual of unexplained signal can be tolerated. The final arbiter is the quality of the image, which is the content of the model domain. Deblending, on the other hand, comes relatively early in the processing workflow, and, unlike the case of migration, it is usually important to preserve as much of the signal as possible for the next stage of processing. What the deblended signal looks like in the model/transform domain (i.e. the focal domain coefficients) is less important. Therefore the range limitations of propagation/focal operators may not be of concern when migrating, they certainly are a cause of concern when deblending. Generally this has to be kept in mind whenever a deblending method employs some type of seismic forward forward modeling that cannot reproduce the entirety of the desired deblended signal. Events that are not faithfully reconstructed, e.g. smeared direct waves in the case of the focal transform, may create crosstalk in the deblended result that might be difficult to remove by subsequent processing.

■ 7.1.3 Balancing transform contributions

When creating a dictionary as a concatenation of multiple transforms, the relative ℓ_2-norm of the atoms will affect the result, as it will influence transform domain amplitudes and consequently which coefficients are kept after sparse inversion. A first measure to combat this is to normalize all atoms to have unit ℓ_2-norm. An overall scaling might be needed for further control, but choosing an optimal value for it is not straightforward.

■ 7.1.4 Shape of the focal transform atoms

As discussed in chapter 3, each atom of the dictionary defined by the focal transform has a hyperbolic moveout. This makes the focal transform good at capturing the curved near-offset part of events as well as diffractions. It struggles, however, to explain very local features, such as sudden changes in amplitude due to anomalies in the gain settings of hydrophones, a situation which sometimes occurs in marine data. These issues would need to be resolved before deblending, as a separate step. The atoms themselves in principle extend spatially to infinity, but geometric spreading somewhat limits in practice this extent, making the atom more localized. Still, very local features can be difficult to explain as the composition of few atoms. This is an area where having a concept of scale, such as that shared by the wavelet and curvelet family of transforms, can be beneficial. It should be noted, however, that the capability of explaining very local features is a double-edged sword, as blending noise from a time delay code can be also interpreted as a set of 'very local amplitude features'.

■ 7.1.5 Sparse representation of seismic data

Complex seismic data can be difficult to explain with only a few atoms from a dictionary of simple shapes. Because of this, perfect reconstruction from solving a BPDN problem is not guaranteed. This manifests as having solutions with blending noise that have a lower ℓ_1-norm than the desired, noise-free solution. Despite this, however, sparsity-based optimization can still be effective in suppressing the amount of blending noise that leaks into the result, making it a useful tool for deblending.

■ 7.1.6 Computational complexity of the focal transform

Computing the double focal transform can require significant computational effort. Assuming a $(n_r \times n_s \times n_f)$ data cube and two focal operator cubes of with dimensions $(n_g \times n_r \times n_f)$ and $(n_s \times n_o \times n_f)$, the number of complex multiplications required for one focusing step for n_x focal subdomain is given by

$$\begin{aligned} &\min\{(n_g n_r n_s + n_g n_s n_o) n_f n_x, (n_g n_r n_o + n_r n_s n_o) n_f n_x\} \equiv \\ &\min\{(n_r + n_o) n_g n_s n_f n_x, (n_g + n_s) n_r n_o n_f n_x\}, \end{aligned} \tag{7.1.1}$$

where n_s (n_r) is the number of unique source (receiver) coordinates, n_g is the number of unique focal grid points, n_o is the number of local subsurface offset coordinates per grid point and n_f is the number of positive temporal frequencies. The reason for taking the minimum between two numbers in (7.1.1) is that applying the double focal transform involves two matrix multiplications per frequency slice, and the order in which they are performed matters when counting the number of elementary operations. Usually one of the two options is preferable. Note that complex additions are ignored here for simplicity.

For the 2D line example in chapter 6.1, $n_g = n_o = n_s = n_r = n$ and the number of complex multiplications becomes $2n^3 n_f n_x$, which implies that if the subsurface focal grid points and local offsets for each grid point coincide (ignoring depth) with the surface acquisition coordinates, the number of operations scales with the third power of the survey size. Although this estimation is rather pessimistic, as n_s and n_r will differ significantly for realistic acquisitions, it indicates that the scaling with respect to survey size can be quite steeper than linear. Focusing and defocusing operations typically need to happen hundreds of times within the deblending inversion. The single-sided version of the transform involves one matrix multiplication per frequency slice per focal subdomain, and thus needs roughly half the number of operations.

Using subsets of the data, as described in chapter 4, can help, as can limiting the number of the local subsurface offset points. Using subsets, however, comes at the cost of truncating the aperture of the Rayleigh integral, possibly leading to worse focusing. Limiting the number of local offset traces too much also runs the risk of leading to the focal transform being unable to satisfactorily explain AVO and other amplitude effects.

■ 7.1.7 General conclusions

Given the above observations, focal deblending as described here is best suited for separating blended wavefields that do not contain strong out-of-plane events and preferably originate from a relatively flat-layered subsurface. In other words, it is best suited for surveys over a subsurface with a character that is closer to 1.5D rather than 3D, unless a more sophisticated focusing mechanism is adopted. Handling realistic survey sizes remains a challenging problem, even when working with subsets. A potential remedy is to exploit the fact that the focal transform is mostly advantageous in explaining the curved near-offset part of the data and limit its contributions to that area. While not optimal, as mentioned earlier, it is a pragmatic compromise. A different transform, such as windowed Fourier, Radon or curvelets should handle the far offsets and what is left unexplained by the focal transform. In fact, using a hybrid transform, or some other way of dealing with the range limitations of the focal transform, is generally required for minimizing blending-noise leakage in realistic datasets.

For low blending factors, the focal transform based on homogeneous velocities can suppress most blending noise, however its ability to do so depends on how compressed the representation of the ideal deblending result is in the focal domain. For this reason, blending factors above 3 are likely to require progressively more complicated propagation physics for achieving separation of acceptable quality, the bar for acceptability being, of course, case-specific.

7.2 Suggestions for future research

■ 7.2.1 Time-domain implementation of the focal transform

The implementation of the focal transform as discussed in this thesis is in the FX domain. Its main advantage over the FK domain is that it does not make the assumption of a wavefield that is uniformly sampled in the $x-$ and $y-$dimensions. Both the FX and FK domains have an advantage over the TX domain, which is that the convolutions/correlations along the time axis implied by the Rayleigh integral simplify to complex number multiplications. There is a drawback, however. As sparsity is to be exploited in the TX form of the focal subdomains, forward/inverse FFTs need to be calculated for every trace and every inversion iteration.

A way to add more flexibility to focal deblending and reduce somewhat the computational burden could be to work entirely in the TX domain. For this approach to be less computationally expensive than working in the FX domain, applying the blending/focal operators to data should take the form of convolutions/correlations with short kernels. A way to achieve this is to use windowed least-squares FIR filter design to approximate the frequency response of the operators, similar to the techniques mentioned by Thorbecke et al. [2004], and references therein. Taking advantage of the fact that seismic data is often relatively oversampled in the time coordinate, the designed filters could trade accuracy in the high frequencies for filter shortness in the time domain. Such an implementation would allow:

- Avoiding expensive FFT evaluations that otherwise need to be calculated in every iteration.
- ‘Small’ focal subdomains that have fewer time samples than the input data. Then, each focal subdomain could explain only a certain time portion of the data, further reducing the amount of needed calculations. The memory requirements for storing the focal subdomains would also reduce.
- Taking advantage of the impulsive nature of strong blending noise and implement the Rayleigh integral using trimmed inner products or other such nonlinear outlier-suppressing techniques [Bednar and Watt, 1984; Chen et al., 2013]. This would yield focal subdomains, and therefore a gradient, which would technically be inaccurate, but also less contaminated by blending noise. Focusing is linked to gradient calculation, and a ‘cleaner’ gradient could lead to a more desirable solution, at the cost of extra computation and possible convergence difficulties due to the fact that the focal operators would effectively change in every iteration.

■ 7.2.2 Incorporate more accurate velocity models

The focal operators used in this thesis are constructed assuming propagation from the surface to a chosen depth level in a homogeneous acoustic medium. This model could be partially extended to handle inhomogeneities in the medium by replacing it with a smooth version and then using that to construct traveltime tables via the eikonal equation. These traveltimes could be used to construct more accurate operators that achieve better focusing, at least for primary reflections. Having the capability to use more accurate velocity models also enables a more thorough study on the influence of velocity errors in the result. A possible drawback is that the focal transform atoms could have a complicated moveout, raising the question of whether it would be possible to represent sparsely seismic events that do not match the complicated moveout of the atoms.

■ 7.2.3 Use concepts from artificial intelligence (AI)

The remarkable successes of AI-based techniques in various fields of science and engineering has sparked intense research in using these techniques for effective seismic data processing. Some of the problems that have been studied include first-break picking [Yuan et al., 2018], denoising [Yu et al., 2019], removal of multiples [Li, 2020], velocity model building [Yang and Ma, 2019; Fabien-Ouellet and Sarkar, 2020; He and Wang, 2021] and migration [Vamaraju and Sen, 2019]. The relevant aspect of AI techniques here is the capability of generating maps that encode complex relationships between input data and desired results, for which we either do not have explicit models, or these models are inexact and/or computationally expensive to use. It is possible that, in the near future, AI-based techniques will be a standard tool of the geophysicist's toolbox.

A possible way to borrow elements from AI methods, is to adopt a deep image prior-type approach [Lempitsky et al., 2018] that incorporates the blending operator, which is known exactly, i.e.,

$$\min_{\theta} \left\{ \left|\left| \mathbf{p}_{\mathrm{ps}} - \boldsymbol{\Psi}^{\mathrm{T}} \boldsymbol{\Psi} \mathbf{d}(\mathbf{p}_{\mathrm{ps}}, \theta) \right|\right|_2^2 \right\}, \tag{7.2.2}$$

where $\mathbf{d}(\cdot, \theta)$ is a nonlinear function implemented as an encoder/decoder neural network (NN) with weight parameters θ. The purpose of this function is to map pseudo-deblended data to clean, deblended data. $\boldsymbol{\Psi}$ is the deblending operator and $\mathbf{p}_{\mathrm{ps}}$ is the pseudo-deblended input data. An advantage of working with pseudo-deblended rather than blended data is that the input and output of the denoising function has the same size, which can make the implementation of the NN easier. It is also often the way field blended data are stored.

Solving the optimization problem yields parameters θ such that $\mathbf{d}(\mathbf{p}_{\mathrm{ps}}, \theta)$, which produces the deblended data, satisfies the blending equation, which here serves as the loss function. The philosophy of this approach is to retain the structure of a usual inversion-based deblending algorithm, but replace what would be a sparsifying transform with an encoder/decoder NN, with hidden layers that get progressively smaller in the encoder section and which progressively expand in the decoder section. The hope here is that the NN will extract a compressed version of the data that explains mostly coherent signal. Then the corresponding blending noise would be generated by the $\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}$ operator.

Unlike many conventional approaches, including the one investigated in this thesis, there is no prescribed dictionary of elementary atoms and determining the denoising NN is an entirely data-driven process. Additionally, the different NN layers operate on progressively more ‘summarized’ versions of the data, giving scale-like qualities to the algorithm.

Training is not explicitly mentioned here, but it could take the form of pre-fitting parameters θ using a number of training datasets and then keeping the parameters corresponding to the first few layers fixed during the inversion phase, when 7.2.2 is solved. This technique is commonly referred to as transfer learning and, in this case, it allows bringing ’knowledge’ from other datasets into the deblending process.

A

Focal deblending as a BPDN problem

A.1 Vectorizing the problem

The objective of this appendix is to show how the focal deblending problem

$$\min_{\substack{\hat{\mathbf{X}}_m^D(z_m,\, z_m)\\ m=1,\ldots,M}} \left\{ \sum_t \sum_{m=1}^{M} \left|\left| \hat{\mathbf{X}}_m^D(z_m,\, z_m) \right|\right|_{\mathrm{S}} \right\} \quad \text{subject to}$$

$$\sum_\omega \left|\left| \mathbf{P}_{\mathrm{bl}}(z_0,\, z_0) - \sum_{m=1}^{M} \mathbf{W}_0^-(z_m,\, z_0)\mathbf{X}_m^D(z_m,\, z_m)\mathbf{W}_m^+(z_m,\, z_0)\mathbf{\Gamma}(z_0) \right|\right|_{\mathrm{F}} \leq \sigma, \quad \text{(A.1.1)}$$

for time slices and frequency slices evaluated at

$$\begin{aligned} t_n &= n\delta t, & n &= 0, 1, \ldots, n_t, \\ \omega_{n'} &= n'\delta\omega = n'\frac{2\pi}{n_t\delta t}, & n' &= 0, 1, \ldots, n_t, \end{aligned} \quad \text{(A.1.2)}$$

can be transformed to the canonical, vectorized form of BPDN [van den Berg and Friedlander, 2008],

$$\min_{\mathbf{x}} \{||\mathbf{x}||_1\} \quad \text{subject to} \quad ||\mathbf{p}_{\mathrm{bl}} - \mathbf{A}\mathbf{x}||_2 \leq \sigma. \quad \text{(A.1.3)}$$

Here $\mathbf{A} = \mathbf{\Psi}\mathbf{\Phi}$, i.e., the composition of an inverse focal transform operator ($\mathbf{\Phi}$) and a blending operator ($\mathbf{\Psi}$), which operate entirely in the time domain. The vectors $\mathbf{p}_{\mathrm{bl}}$ and $\mathbf{x}$ are the time-domain vectorized blended data and concatenated focal subdomain data respectively. Equation (A.1.3) is the vectorized form of (3.5.34). It is also related to (3.3.16) when $\mathbf{\Gamma}(z_0) = \mathbf{I}$, and thus $\mathbf{\Psi} = \mathbf{I}$.

The first step in this process is to rewrite the multiplications of frequency slices in (A.1.1) as discrete summations. For the following discussion the notation $[\mathbf{A}]_{i,j}$ will be used for the i-th row, j-th column element of a matrix $\mathbf{A}$, and $[\mathbf{x}]_i$ for the i-th element of a vector $\mathbf{x}$. From chapters 2 and 3 we have that

$$\begin{aligned}
p_{\mathrm{bl}}(\mathbf{r}_q,\, b;\, \omega) &= [\mathbf{P}_{\mathrm{bl}}(z_0,\, z_0)]_{q,b},\\
w_m^-(\mathbf{r}_q,\, \mathbf{u}_l';\, \omega) &= [\mathbf{W}_m^-(z_0,\, z_m)]_{q,l},\\
x_m^D(\mathbf{u}_l',\, \mathbf{u}_k;\, \omega) &= [\mathbf{X}_m^D(z_m,\, z_m)]_{l,k},\\
x_m^D(\mathbf{u}_l',\, \mathbf{u}_k;\, t) &= [\hat{\mathbf{X}}_m^D(z_m,\, z_m)]_{l,k},\\
w_m^+(\mathbf{u}_k,\, \mathbf{s}_i;\, \omega) &= [\mathbf{W}_m^+(z_m,\, z_0)]_{k,i},\\
\gamma(\mathbf{s}_i,\, b;\, \omega) &= [\mathbf{\Gamma}(z_0)]_{i,b}.
\end{aligned} \tag{A.1.4}$$

Very often pairs of coordinates will need to be matched to corresponding traces. For this purpose the following pair-to-flat index maps of the form $(i_1, i_2) \to \kappa$ are introduced:

- $(q, b) \to \xi$, from the q-th receiver in the b-th blended shot gather to the ξ-th blended data trace.
- $(l, k) \to \nu$, from the l-th and k-th point of the focal grid to the ν-th trace of a focal subdomain.
- $(q, i) \to \mu$, from the q-th receiver and i-th shot location to the μ-th unblended data trace.

In order to simplify the summation limits, it is assumed that each of the n_s unblended shot records and each of the b blended shot records has n_r traces. Similarly, for each of the n_u grid points, the focal transform will be evaluated at n_u local subsurface offsets.

Using (A.1.4) the problem (A.1.1) can be rewritten using summations over individual elements as

$$\min_{\substack{x_m^D(\mathbf{u}_l',\, \mathbf{u}_k;\, t_n)\\ m=1,\ldots,M\\ l,k=1,\ldots,n_u\\ n=0,\ldots,n_t-1}} \left\{ \sum_{n=0}^{n_t-1} \sum_{m=1}^{M} \sum_{l=1}^{n_u} \sum_{k=1}^{n_u} |x_m^D(\mathbf{u}_l',\, \mathbf{u}_k;\, t_n)| \right\} \quad \text{s.t.}$$

$$\left(\sum_{n'=0}^{n_t-1} \sum_{q=1}^{n_r} \sum_{b=1}^{n_{\mathrm{bl}}} \left| p_{\mathrm{bl}}(\mathbf{r}_q,\, b;\, \omega_{n'}) - \sum_{i=1}^{n_s} \left(\sum_{m=1}^{M} \sum_{l=1}^{n_u} \sum_{k=1}^{n_u} w_m^-(\mathbf{r}_q,\, \mathbf{u}_l';\, \omega_{n'}) \times \right.\right.\right.$$
$$\left.\left.\left. \times\, x_m^D(\mathbf{u}_l',\, \mathbf{u}_k;\, \omega_{n'}) w_m^+(\mathbf{u}_k,\, \mathbf{s}_i;\, \omega_{n'}) \right) \gamma(\mathbf{s}_i,\, b;\, \omega_{n'}) \right|^2 \right)^{\frac{1}{2}} \leq \sigma. \tag{A.1.5}$$

To make expressions more concise, the elements of the two focal operators are fused together in

$$\phi_m(\mathbf{r}_q, \mathbf{u}'_l, \mathbf{u}_k, \mathbf{s}_i; \omega) = w^-_m(\mathbf{r}_q,\, \mathbf{u}'_l;\, \omega) w^+_m(\mathbf{u}_k,\, \mathbf{s}_i;\, \omega). \tag{A.1.6}$$

The next step is to transform (A.1.5) to the time domain. This can be done with the aid of the Parseval and convolution theorems for the DFT. Then, the constraint part of (A.1.5) can be rewritten as

$$\Bigg(\sum_{n=0}^{n_t-1} \sum_{q=1}^{n_r} \sum_{b=1}^{n_{\text{bl}}} \Bigg| p_{\text{bl}}(\mathbf{r}_q,\, b;\, t_n) - \sum_{i=1}^{n_s} \sum_{m=1}^{M} \sum_{l=1}^{n_u} \sum_{k=1}^{n_u} \sum_{n'=0}^{n_t-1} \sum_{n''=0}^{n_t-1} \\ \gamma(\mathbf{s}_i,\, b;\, t_{\text{mod}(n-n',n_t)}) \phi_m(\mathbf{r}_q, \mathbf{u}'_l, \mathbf{u}_k, \mathbf{s}_i; t_{\text{mod}(n'-n'',n_t)}) \times \\ \times\, x^D_m(\mathbf{u}'_l,\, \mathbf{u}_k;\, t_{n''}) \Bigg|^2 \Bigg)^{\frac{1}{2}} \leq \sigma. \tag{A.1.7}$$

Element-by-element multiplication of frequency samples of the operators and the data turn into circular convolutions in the time domain.

The vectorization process starts by assembling time samples pertaining to the same blended trace into vectors $\mathbf{p}^{(\xi)}_{\text{bl}}$. Similarly, time samples of focal subdomain traces are assembled into vectors $\mathbf{x}^{(\nu)}_m$:

$$\begin{aligned} \left[\mathbf{p}^{(\xi)}_{\text{bl}}\right]_n &= p_{\text{bl}}(\mathbf{r}_{q(\xi)},\, b(\xi);\, t_n), \\ \left[\mathbf{x}^{(\nu)}_m\right]_{n''} &= x^D_m(\mathbf{u}'_{l(\nu)},\, \mathbf{u}_{k(\nu)};\, t_{n''}), \\ n, n'' &= 0, 1, \ldots, n_t - 1. \end{aligned} \tag{A.1.8}$$

A matrix $\mathbf{\Phi}^{(\mu,\nu)}_m$ is defined such that $\mathbf{\Phi}^{(\mu,\nu)}_m \mathbf{x}^{(\nu)}_m$ is the contribution of the ν-th trace of the m-th focal subdomain to the μ-th trace of the estimated deblended data. In a similar fashion $\mathbf{\Psi}^{(\xi,\mu)}$ calculates the contribution of the μ-th unblended data trace to the ξ-th blended trace. These matrices have elements given by

$$\begin{aligned} \left[\mathbf{\Psi}^{(\xi,\mu)}\right]_{n,n'} &= \gamma(\mathbf{s}_{i(\xi)},\, b(\xi);\, t_{\text{mod}(n-n',n_t)}), \\ \left[\mathbf{\Phi}^{(\mu,\nu)}_m\right]_{n',n''} &= \phi_m(\mathbf{r}_{q(\mu)}, \mathbf{u}'_{l(\nu)}, \mathbf{u}_{k(\nu)}, \mathbf{s}_{i(\mu)}; t_{\text{mod}(n'-n'',n_t)}), \\ n, n', n'' &= 0, 1, \ldots, n_t - 1. \end{aligned} \tag{A.1.9}$$

They are both circulant matrices. With the aid of (A.1.8) and (A.1.9), the time-domain form (A.1.7) becomes

$$\min_{\substack{\mathbf{x}^{(\nu)}_m \\ m=1,\ldots,M \\ \nu=1,\ldots,n_u^2}} \left\{ \sum_{m=1}^{M} \sum_{\nu=1}^{n_u^2} \left\| \mathbf{x}^{(\nu)}_m \right\|_1 \right\} \quad \text{subject to}$$

$$\sqrt{\sum_{\xi=1}^{n_r b} \left\| \mathbf{p}_{\mathrm{bl}}^{(\xi)} - \sum_{\mu=1}^{n_s n_r} \mathbf{\Psi}^{(\xi,\mu)} \sum_{m=1}^{M} \sum_{\nu=1}^{n_u^2} \mathbf{\Phi}_m^{(\mu,\nu)} \mathbf{x}_m^{(\nu)} \right\|_2^2} \leq \sigma. \tag{A.1.10}$$

The next level in the vectorization process is to concatenate the ν-th trace of each focal subdomain one after the other, as well as the corresponding matrices $\mathbf{\Phi}_m^{(\mu,\nu)}$ into

$$\begin{aligned} \mathbf{x}^{(\nu)} &= \left[\mathbf{x}_1^{(\nu)\mathrm{T}}, \mathbf{x}_2^{(\nu)\mathrm{T}}, \ldots, \mathbf{x}_M^{(\nu)\mathrm{T}} \right]^{\mathrm{T}}, \\ \mathbf{\Phi}^{(\mu,\nu)} &= \left[\mathbf{\Phi}_1^{(\mu,\nu)}, \mathbf{\Phi}_2^{(\mu,\nu)}, \ldots, \mathbf{\Phi}_M^{(\mu,\nu)} \right]. \end{aligned} \tag{A.1.11}$$

Substituting in (A.1.10) yields a yet more compact form,

$$\begin{gathered} \min_{\substack{\mathbf{x}_m^{(\nu)} \\ \nu=1,\ldots,n_u^2}} \left\{ \sum_{\nu=1}^{n_u^2} \left\| \mathbf{x}^{(\nu)} \right\|_1 \right\} \quad \text{subject to} \\ \sqrt{\sum_{\xi=1}^{n_r b} \left\| \mathbf{p}_{\mathrm{bl}}^{(\xi)} - \sum_{\mu=1}^{n_s n_r} \sum_{\nu=1}^{n_u^2} \mathbf{\Psi}^{(\xi,\mu)} \mathbf{\Phi}^{(\mu,\nu)} \mathbf{x}^{(\nu)} \right\|_2^2} \leq \sigma. \end{gathered} \tag{A.1.12}$$

The last step in the vectorization involves concatenating the $n_r b$ blended traces, as well as the $M n_u^2$ focal domain traces into vectors

$$\begin{aligned} \mathbf{p}_{\mathrm{bl}} &= \left[\mathbf{p}_{\mathrm{bl}}^{(1)\mathrm{T}}, \mathbf{p}_{\mathrm{bl}}^{(2)\mathrm{T}}, \ldots, \mathbf{p}_{\mathrm{bl}}^{(n_s n_r)\mathrm{T}} \right]^{\mathrm{T}}, \\ \mathbf{x} &= \left[\mathbf{x}^{(1)\mathrm{T}}, \mathbf{x}^{(2)\mathrm{T}}, \ldots, \mathbf{x}^{(n_u^2)\mathrm{T}} \right]^{\mathrm{T}}. \end{aligned} \tag{A.1.13}$$

The final operator matrices are also formed as follows:

$$\begin{aligned} \mathbf{\Phi} &= \begin{bmatrix} \mathbf{\Phi}^{(1,1)} & \mathbf{\Phi}^{(1,2)} & \ldots & \mathbf{\Phi}^{(1,n_u^2)} \\ \mathbf{\Phi}^{(2,1)} & \mathbf{\Phi}^{(2,2)} & \ldots & \mathbf{\Phi}^{(2,n_u^2)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Phi}^{(n_s n_r,1)} & \mathbf{\Phi}^{(n_s n_r,2)} & \ldots & \mathbf{\Phi}^{(n_s n_r,n_u^2)} \end{bmatrix}, \\ \mathbf{\Psi} &= \begin{bmatrix} \mathbf{\Psi}^{(1,1)} & \mathbf{\Psi}^{(1,2)} & \ldots & \mathbf{\Psi}^{(1,n_s n_r)} \\ \mathbf{\Psi}^{(2,1)} & \mathbf{\Psi}^{(2,2)} & \ldots & \mathbf{\Psi}^{(2,n_s n_r)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Psi}^{(n_r b,1)} & \mathbf{\Psi}^{(n_s n_r,2)} & \ldots & \mathbf{\Psi}^{(n_r b,n_s n_r)} \end{bmatrix}. \end{aligned} \tag{A.1.14}$$

Plugging into (A.1.12) yields the canonical BPDN form,

$$\min_{\mathbf{x}} \{||\mathbf{x}||_1\} \quad \text{subject to} \quad ||\mathbf{p}_{\text{bl}} - \boldsymbol{\Psi}\boldsymbol{\Phi}\mathbf{x}||_2 \leq \sigma. \tag{A.1.15}$$

A.2 Time-domain structure of the blending and focal transform operators

So far $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ have been fairly generic. $\boldsymbol{\Phi}$ encodes the double focal transform and $\boldsymbol{\Psi}$ a blending code. By making specific choices for the kind of focal operators and type of blending, it is possible to visualize the structure present in $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$. The combination of 3D focal operators and time-delay blending will be examined below.

■ A.2.1 The elements of $\boldsymbol{\Phi}$

For 3D operators we have from table 3.1 that

$$\begin{aligned} w_m^-(\mathbf{r}_q,\, \mathbf{u}_l';\, \omega) &= \frac{z_m(1 + j\omega\tau_{ql})}{\Delta r_{ql}^3} e^{-j\omega\tau_{ql}}, \\ w_m^+(\mathbf{u}_k,\, \mathbf{s}_i;\, \omega) &= \frac{z_m(1 + j\omega\tau_{ki})}{\Delta r_{ki}^3} e^{-j\omega\tau_{ki}}. \end{aligned} \tag{A.2.16}$$

with

$$\begin{aligned} \tau_{ql} &= \Delta r_{ql}/c_m, \quad \Delta r_{ql} = ||\mathbf{r}_q - \mathbf{u}_l'||_2, \\ \tau_{ki} &= \Delta r_{ki}/c_m, \quad \Delta r_{ki} = ||\mathbf{u}_k - \mathbf{s}_i||_2, \\ \tau_{qlki} &= \tau_{ql} + \tau_{ki}. \end{aligned} \tag{A.2.17}$$

Here z_m and c_m are the depth and velocity associated with the m-th focal subdomain.

Given this choice, $\phi_m(\cdot)$ in the frequency domain takes the explicit form

$$\begin{aligned} \phi_m(\mathbf{r}_q, \mathbf{u}_l', \mathbf{u}_k, \mathbf{s}_i; \omega) &= \left(\frac{z_m(1 + j\omega\tau_{ql})}{\Delta r_{ql}^3} e^{-j\omega\tau_{ql}} \right) \left(\frac{z_m(1 + j\omega\tau_{ki})}{\Delta r_{ki}^3} e^{-j\omega\tau_{ki}} \right) \\ &= \frac{z_m^2}{\Delta r_{ql}^3 \Delta r_{ki}^3} \left[1 + \tau_{qlki}(j\omega) + \tau_{ql}\tau_{ki}(j\omega)^2 \right] e^{-j\omega\tau_{qlki}}. \end{aligned} \tag{A.2.18}$$

Note that the presence of coefficients of the form $(j\omega)^k$ generates time derivatives of the time-shift operator $e^{-j\omega\tau_{qlki}}$. The term that dominates is the one involving the second derivative, as it is related to the far-field component of both focal operators.

The inverse DFT is the applied to get a time-domain expression, taking care to enforce conjugate symmetry between positive and negative frequencies:

$$\begin{aligned}
\phi_m(\mathbf{r}_q, \mathbf{u}'_l, &\mathbf{u}_k, \mathbf{s}_i; t_n) = \\
&\frac{1}{n_t} \sum_{n'=0}^{\lceil n_t/2 \rceil} \phi_m(\mathbf{r}_q, \mathbf{u}'_l, \mathbf{u}_k, \mathbf{s}_i; \omega_{n'}) e^{j\omega_{n'} t_n} \\
&+ \frac{1}{n_t} \sum_{\substack{n'=\lceil n_t/2 \rceil + 1 \\ -\mathrm{mod}(n_t, 2)}}^{n_t - 1} \phi_m(\mathbf{r}_q, \mathbf{u}'_l, \mathbf{u}_k, \mathbf{s}_i; \omega_{n'-n_t}) e^{j\omega_{n'} t_n}.
\end{aligned} \tag{A.2.19}$$

Note that presence of the Nyquist frequency when n_t is even has to treated specially, in order to ensure conjugate symmetry. The choice made here was to exclude it from the sum, which is equivalent to forcing the operator to have a zero value at the Nyquist frequency, i.e.

$$\phi_m(\mathbf{r}_q, \mathbf{u}'_l, \mathbf{u}_k, \mathbf{s}_i; \omega_{\lceil n_t/2 \rceil}) = 0 \quad \text{when } n_t \text{ is even.} \tag{A.2.20}$$

The operator is then less accurate, as its nullspace is modified by this change. Fortunately this is not a problem in practice, as there is usually little spectral content near the Nyquist frequency for the typical sampling rates used in seismic data acquisition.

In the time domain the operator $\phi_m(\mathbf{r}_q, \mathbf{u}'_l, \mathbf{u}_k, \mathbf{s}_i; t)$ takes the form

$$\begin{aligned}
\phi_m(\mathbf{r}_q, \mathbf{u}'_l, \mathbf{u}_k, \mathbf{s}_i; t_n) &= \frac{z_m^2}{\Delta r_{ql}^3 \Delta r_{ki}^3} \times \\
&\times \left[S(t_n - \tau_{qlki}) + \tau_{qlki} S'(t_n - \tau_{qlki}) + \tau_{ql} \tau_{ki} S''(t_n - \tau_{qlki}) \right],
\end{aligned} \tag{A.2.21}$$

where $S_C(t)$, $S'_C(t)$ and $S''_C(t)$ are the periodic sinc function, its first and its second derivative, respectively. They are defined as

$$\begin{aligned}
S_C(t) &= \sin(\alpha t))/(n_t \sin(\beta t)), \\
S'_C(t) &= (\alpha \cot(\alpha t) - \beta \cot(\beta t)) S_C(t), \\
S''_C(t) &= (\beta^2 - \alpha^2) S_C(t) - 2\beta \cot(\beta t) S'_C(t),
\end{aligned} \tag{A.2.22}$$

for $t \neq k n_t \delta t$, $k \in \mathbb{Z}$, with

$$\alpha = \frac{(n_t - 1 + \mathrm{mod}(n_t, 2))\pi}{n_t \delta t} \quad \text{and} \quad \beta = \frac{\pi}{n_t \delta t}. \tag{A.2.23}$$

For the special case $t = k n_t \delta t$, they take the values

$$S_C(k n_t \delta t) = \frac{\alpha}{\beta n_t}, \quad S'_C(k n_t \delta t) = 0, \quad S''_C(k n_t \delta t) = \frac{\alpha(\beta^2 - \alpha^2)}{3\beta n_t}. \tag{A.2.24}$$

These expressions are valid for both even and odd n_t, when the Nyquist frequency is excluded in the even n_t case.

An insight from (A.2.21) is that the cumulative effect of the two focal operators in the inverse focal transform mode is to add an amount of offset-dependent periodic time shift to each focal subdomain trace, followed by taking a linear combination of the resulting wavefield and its first and second derivatives. The presence of time derivatives makes the focal transform slightly different from Radon-type transforms.

■ A.2.2 The elements of $\boldsymbol{\Psi}$

For a time delay blending code, we have from (2.2.9) that

$$\gamma(\mathbf{s}_i,\, b;\, \omega) = a(\mathbf{s}_i,\, b) e^{-j\omega\tau_d(\mathbf{s}_i,\, b)}, \tag{A.2.25}$$

with

$$a(\mathbf{s}_i,\, b) = \begin{cases} 1, & \text{if the } i\text{-th shot contributes to the } b\text{-th blended gather,} \\ 0, & \text{otherwise,} \end{cases}$$

$$\tau_d(\mathbf{s}_i,\, b) = \begin{cases} \tau_{d,i}, & \text{the blending code delay associated with the } i\text{-th shot, if} \\ & \text{that shot contributes to the } b\text{-th blended gather,} \\ 0, & \text{otherwise.} \end{cases}$$

Applying the inverse DFT as before, again enforcing conjugate symmetry,

$$\begin{aligned} \gamma(\mathbf{s}_i,\, b;\, t_n) =& \frac{1}{n_t} \sum_{n'=0}^{\lceil n_t/2 \rceil} \gamma(\mathbf{s}_i,\, b;\, \omega_{n'}) e^{j\omega_{n'} t_n} \\ &+ \frac{1}{n_t} \sum_{\substack{n'=\lceil n_t/2 \rceil + 1 \\ -\mathrm{mod}(n_t,2)}}^{n_t - 1} \gamma(\mathbf{s}_i,\, b;\, \omega_{n'-n_t}) e^{j\omega_{n'} t_n} \\ =& a(\mathbf{s}_i,\, b) S_C(t_n - \tau_d(\mathbf{s}_i,\, b)). \end{aligned} \tag{A.2.26}$$

Here the effect of $\gamma(\cdot)$ is to simply apply a periodic shift to the trace, for calculating its contribution to a blended trace, provided that $a(\mathbf{s}_i,\, b) = 1$.

■ A.2.3 $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ visualized

Given explicit expressions for $\phi_m(\cdot)$ and $\gamma(\cdot)$, the matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ can be constructed as discussed previously. An example of a matrix $\boldsymbol{\Phi}$ for two focal subdomains, each with two focal traces can be seen in figure A.1b. It produces three unblended traces, which can be blended by applying $\boldsymbol{\Psi}$, seen in figure A.1a. This produces two output traces, of which the first is blended and the second simply time-shifted.

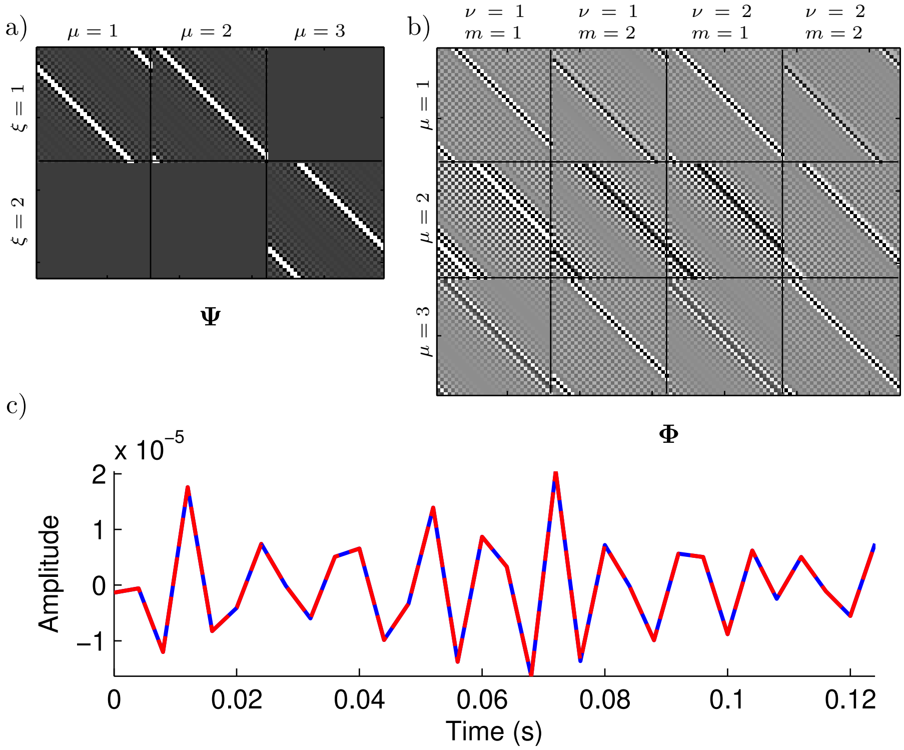


Figure A.1: *Visualization of a) the matrix* $\mathbf{\Psi}$*; b) the matrix* $\mathbf{\Phi}$*; c) a blended trace produced by carrying out the operations in the frequency (blue) and the time domain (red).*

Several observations can be made regarding the structure of the matrices, the first of which is that they consist of circulant blocks. This is to be expected, as they need to implement circular convolutions in the time domain. A second observation is that although the individual blocks of $\mathbf{\Phi}$ and the nonzero blocks of $\mathbf{\Psi}$ are full matrices, most significant contributions come from a shifted diagonal band. This agrees with the intuition that time-shifting and differentiation are (in the limit $\delta t \rightarrow 0$) local operations. The 'mostly local' character of these operations enables the possibility of an alternative, time domain implementation of the focal transform, which could approximate these operators with short filters (at the expense of frequency response fidelity) and involve time-domain denoising when deblending. This idea is discussed in section 7.2.

Forming matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ explicitly can be useful for building intuition on how they act on data. For an actual software implementation of focal deblending however, constructing them explicitly is impractical and in most cases impossible, as they tend to be huge, even for small problems with a few thousand traces. For the configuration examined in this appendix, their sizes are

$$\begin{aligned} (n_s n_r n_t \times M n_u^2 n_t) &\quad \text{for } \boldsymbol{\Phi}, \\ (n_r n_t b \times n_s n_r n_t) &\quad \text{for } \boldsymbol{\Psi}. \end{aligned} \tag{A.2.27}$$

Even for a modest sizes, e.g. $n_s = n_r = n_t = n_u = 1000$, these matrices would have billions of rows and columns, which is prohibitive both in terms of storage and computation. An implicit, matrix-free implementation is necessary in practice.

B

SPGL1 for focal deblending

An integral part of the deblending method explored this thesis is sparse optimization. More specifically, the forward focal transform (problem 3.3.18) as well as deblending (problem 3.5.36) were formulated as BPDN problems that have the general form,

$$\mathbf{x}_\sigma = \arg\min_{\mathbf{x}} \{||\mathbf{x}||_1\} \text{ subject to } ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2 \leq \sigma. \tag{B.0.1}$$

An advantage of the BPDN formulation is that its parameter σ controls the maximum amount of tolerated data misfit, which is usually easier to set than directly controlling the amount of sparsity of the solution, as happens in the LASSO or OMP formulation.

SPGL1 is matrix-free numerical solver for the BPDN problem (B.0.1), proposed by van den Berg and Friedlander [2008]. Often matrix-vector products $\mathbf{A}\mathbf{x}$ and $\mathbf{A}\mathbf{r}^{\mathrm{T}}$ can be more efficiently implemented in an algorithmic fashion, exploiting some special structure in $\mathbf{A}$. SPGL1 allows us to take advantage of this. What follows in the rest of this appendix is not original work, but a rather short description of SPGL1, included here for convenience. The reader can refer to van den Berg and Friedlander [2008, 2011] for a more in-depth coverage of the algorithm and its properties.

SPGL1 works by solving a series of LASSO problems

$$\mathbf{x}_\tau = \arg\min_{\mathbf{x}} \{||\mathbf{b} - \mathbf{A}\mathbf{x}||_2\} \text{ subject to } ||\mathbf{x}||_1 \leq \tau, \tag{B.0.2}$$

for different values of the parameter τ. Note that the LASSO and the BPDN problems are related, in the sense that what constitutes the objective function for one becomes part of the constraint in the other. SPGL1 solver identifies the smallest value of τ such that solving (B.0.2) yields a result $\mathbf{x}_\tau$ that satisfies $||\mathbf{b} - \mathbf{A}\mathbf{x}_\tau||_2 \leq \sigma$.

Seen from a geometric point of view, SPGL1 tries to find which is the smallest possible radius of an ℓ_1-ball centered at the origin can have while touching a point in the set $\{\mathbf{x}_\tau : ||\mathbf{b} - \mathbf{A}\mathbf{x}_\tau||_2 \leq \sigma\}$.

To discover the optimal value for τ, SPGL1 samples the Pareto curve

$$\phi(\tau) = ||\mathbf{b} - \mathbf{A}\mathbf{x}_\tau||_2 = ||\mathbf{r}_\tau||_2 \,, \tag{B.0.3}$$

where $\mathbf{x}_\tau$ is the solution of (B.0.2), for a given value of τ. The Pareto curve is a manifestation of the tradeoff between $||\mathbf{x}_\tau||_1$ and $||\mathbf{r}_\tau||_2$. As one would intuitively expect, $\phi(\tau)$ is non-increasing, and greater value for τ will generally lead to a reduction of the residual, up to the point when either it becomes zero (and the solution is then a basis pursuit solution), or it reaches some minimum possible value (when $\mathbf{b} \notin \text{range}(\mathbf{A})$). A remarkable fact that is proven by van den Berg and Friedlander [2008] is that evaluating $\phi(\tau)$ by solving (B.0.2) allows one to evaluate the derivative $\phi'(\tau)$ at the same point as a byproduct:

$$\phi'(\tau) = \frac{||\mathbf{A}^{\mathrm{T}}\mathbf{r}_\tau||_\infty}{||\mathbf{r}_\tau||_2}. \tag{B.0.4}$$

It is also shown by van den Berg and Friedlander [2008] that $\phi(\tau)$ is continuously differentiable. These two facts allow for using the Newton root-finding method for discovering τ_σ and corresponding $\mathbf{x}_\sigma$ such that

$$\phi(\tau_\sigma) = \sigma. \tag{B.0.5}$$

Then $\mathbf{x}_\sigma$ satisfies (B.0.1), which solves the original BPDN problem. The values $\tau^{[0]}, \tau^{[1]}, \ldots$ at which the Pareto curve (figure B.1) is sampled are determined by the Newton iteration

$$\tau^{[k+1]} = \tau^{[k]} - \frac{\phi(\tau) - \sigma}{\phi'(\tau)}. \tag{B.0.6}$$

Note that using the Pareto curve in the manner discussed here is not only applicable to basis pursuit, but can be generalized to handle a broader class of problems [van den Berg and Friedlander, 2011].

SPGL1 can be conceptually divided[1] in 'outer loop iterations', in which τ is modified as per (B.0.6), and 'inner loop iterations', in which a LASSO problem is solved. For the latter, a spectral projected gradient iterative solver is used. Pseudocode descriptions of these two parts can be found in van den Berg and Friedlander [2008] and van den Berg and Friedlander [2011] respectively. They are also listed here as algorithms 1 and 2, with minor modifications to accommodate for the focal deblending application considered in this thesis.

---

[1] The actual implementation [van den Berg and Friedlander, 2007] merges these two loops in one, using a set of criteria to determine when the ℓ_1-ball radius is to be modified.

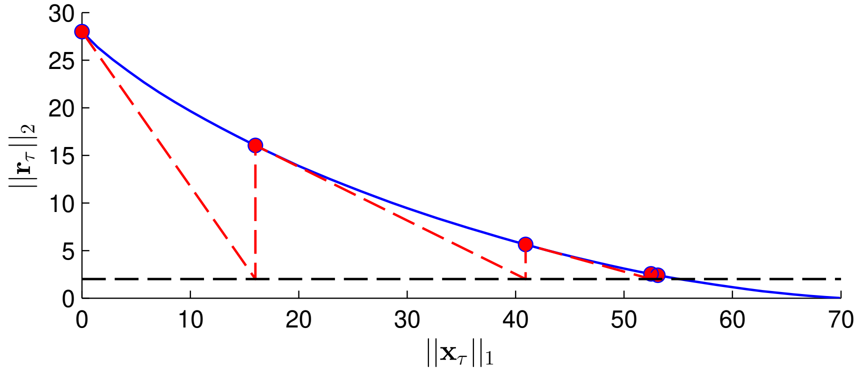


Figure B.1: *Pareto curve for a toy example, similar to figure 2.1 in van den Berg and Friedlander [2008]. Blue line: the Pareto curve; red line: successive estimates of the root location via the Newton method; black line: the noise threshold* σ*.*

The spectral projected gradient solver needs to have a way of projecting some point $\mathbf{c}$ onto the ℓ_1-ball of radius τ, or, in other words, a way of solving the problem

$$\textsc{proj}(\mathbf{c},\tau) = \arg\min_{\mathbf{x}} \{||\mathbf{c}-\mathbf{x}||_2\} \text{ subject to } ||\mathbf{x}||_1 \leq \tau. \tag{B.0.7}$$

An $O(n \log n)$ complexity method based on a binomial heap data structure is proposed for this purpose by van den Berg and Friedlander [2008]. Although convenient for single-core implementations, this approach is challenging to parallelize, because of the heap structure employed. As projection is third the most expensive operation after application of $\mathbf{A}$ and $\mathbf{A}^\mathrm{T}$, it is important to also parallelize for large field data processing. Fortunately, it is possible to calculate the projection in other ways, such as by using the active set method of Michelot [1986], as is proposed by Sun and Verschuur [2018].

The user has to provide an implementation of $\mathbf{A}$ and $\mathbf{A}^\mathrm{T}$, which for focal deblending breaks down to four operations:

- BLEND()/PSDEBLEND(), implementing the blending operation and its adjoint,
- DEFOCUS()/FOCUS(), implementing the inverse focal transform and its adjoint.

These operators need not be defined in explicit matrix form. For deblending, BLEND(DEFOCUS($\cdot$)) and FOCUS(PSDEBLEND($\cdot$)) implement $\mathbf{A}$ and $\mathbf{A}^\mathrm{T}$, respectively.

Algorithm 1 SPGL1 solver for BPDN

```
Input
    $\mathbf{p}_{\mathrm{bl}}$    vectorized blended data
    $\sigma$    target $\ell_2$-norm of the data misfit
    $\delta$    maximum allowed duality gap
    $\epsilon$    data misfit tolerance
Output
    $\mathbf{p}_{\mathrm{debl}}$    vectorized deblended data

function SPGL1($\mathbf{p}_{\mathrm{bl}}, \sigma, \delta, \epsilon$)
    $\mathbf{x}_0 \leftarrow \mathbf{0}$, $\mathbf{r}_0 \leftarrow \mathbf{p}_{\mathrm{bl}}$, $\tau_0 \leftarrow 0$, $k \leftarrow 1$
    while $\left| ||\mathbf{r}_{k-1}||_2^2 - \sigma \right| > \epsilon$ do
        $\mathbf{x}_k, \mathbf{r}_k, \mathbf{g}_k \leftarrow$ SPGLASSO($\mathbf{p}_{\mathrm{bl}}, \mathbf{x}_{k-1}, \tau_{k-1}, \delta$)
        $\phi \leftarrow ||\mathbf{r}_k||_2$, $\phi' \leftarrow -\,||\mathbf{g}_k||_\infty \,/\, ||\mathbf{r}_k||_2$
        $\tau_k \leftarrow \tau_{k-1} - (\phi - \sigma)/\phi'$
        $k \leftarrow k + 1$
    end while
    return $\mathbf{p}_{\mathrm{debl}} \leftarrow$ DEFOCUS($\mathbf{x}_k$)
end function
```

Algorithm 2 Spectral projected gradient solver for LASSO

```
Input
    $\mathbf{p}_{\mathrm{bl}}$    vectorized blended data
    $\mathbf{x}$    vectorized focal subdomains, initial estimate
    $\tau$    $\ell_1$-ball radius
    $\delta$    maximum allowed duality gap
Output
    $\mathbf{x}_\tau$    vectorized focal subdomains, after optimization
    $\mathbf{r}_\tau$    vectorized final residual
    $\mathbf{g}_\tau$    vectorized final gradient
Parameters
    $\alpha_{\min}, \alpha_{\max}$    minimum/maximum step length, $0 < \alpha_{\min} < \alpha_{\max}$
    $\alpha_0$    initial step length, $\alpha_0 \in [\alpha_{\min}, \alpha_{\max}]$
    $\gamma$    sufficient descent parameter, $\gamma \in (0, 1)$
    $M$    line search history length, $M > 0$
```

```
function SPGLASSO($\mathbf{p}_{\mathrm{bl}}, \mathbf{x}, \tau, \delta$)
    $\mathbf{x}_0 \leftarrow$ PROJ($\mathbf{x}, \tau$)
    $\mathbf{r}_0 \leftarrow \mathbf{p}_{\mathrm{bl}} -$ BLEND(DEFOCUS($\mathbf{x}_0$))
    $\mathbf{g}_0 \leftarrow -$FOCUS(PSDEBLEND($\mathbf{r}_0$))
    $l \leftarrow 0$
    loop
        $\delta_l \leftarrow ||\mathbf{r}_l||_2 - \left(\mathbf{p}_{\mathrm{bl}}^{\mathrm{T}}\mathbf{r}_l - \tau\, ||\mathbf{g}_l||_\infty\right)$
        if $\delta_l < \delta$ break
        $\alpha \leftarrow \alpha_l$
        loop
            $\bar{\mathbf{x}} \leftarrow$ PROJ($\mathbf{x}_l - \alpha\mathbf{g}_l, \tau$)
            $\bar{\mathbf{r}} \leftarrow \mathbf{p}_{\mathrm{bl}} -$ BLEND(DEFOCUS($\bar{\mathbf{x}}$))
            $r_{\max} \leftarrow \max_{j\in[0,\,\min\{l,M-1\}]}\{||\mathbf{r}_{l-j}||_2^2\}$
            $\bar{\beta} \leftarrow \gamma(\bar{\mathbf{x}} - \mathbf{x}_l)^{\mathrm{T}}\mathbf{g}_l$
            if $||\bar{\mathbf{r}}||_2^2 \leq r_{\max} + \bar{\beta}$ break else $\alpha \leftarrow \alpha/2$
        end loop
        $\mathbf{x}_{l+1} \leftarrow \bar{\mathbf{x}}$
        $\mathbf{r}_{l+1} \leftarrow \bar{\mathbf{r}}$
        $\mathbf{g}_{l+1} \leftarrow -$FOCUS(PSDEBLEND($\mathbf{r}_{l+1}$))
        $\Delta\mathbf{x} \leftarrow \mathbf{x}_{l+1} - \mathbf{x}_l$
        $\Delta\mathbf{g} \leftarrow \mathbf{g}_{l+1} - \mathbf{g}_l$
        if $\Delta\mathbf{x}^{\mathrm{T}}\Delta\mathbf{g} \leq 0$ then
            $\alpha_{l+1} \leftarrow \alpha_{\max}$
        else
            $\alpha_{l+1} \leftarrow \min\left\{\alpha_{\max}, \max\left\{\alpha_{\min}, \frac{\Delta\mathbf{x}^{\mathrm{T}}\Delta\mathbf{x}}{\Delta\mathbf{x}^{\mathrm{T}}\Delta\mathbf{g}}\right\}\right\}$
        end if
        $l \leftarrow l + 1$
    end loop
    return $\mathbf{x}_\tau \leftarrow \mathbf{x}_l$, $\mathbf{r}_\tau \leftarrow \mathbf{r}_l$, $\mathbf{g}_\tau \leftarrow \mathbf{g}_l$
end function
```

C

The blending noise operator

For the following discussion we assume a blending operator $\mathbf{\Gamma}(z_0)$ with the following properties: a) it implements a purely random time delay code, b) each shot participates in one blended shot gather, c) each blended shot gather is produced by blending at least two shot gathers. For every column i of $\mathbf{\Gamma}(z_0)$ we define a set $\mathcal{B}_i$ that holds the indices of the shots that participate in the i-th blended shot gather, i.e.

$$\mathcal{B}_i = \{j : \gamma(\mathbf{s}_j, i; \omega) \neq 0\}, \quad i = 1, 2, \ldots, n_{\mathrm{bl}}, \quad j = 1, 2, \ldots, n_s, \tag{C.0.1}$$

where $\gamma(\mathbf{s}_j, i; \omega)$ is as defined in (2.2.9), $\mathbf{s}_j$ is the j-th shot location, n_{bl} is the number of blended shot gathers and n_s the number of shots. Property b) then means that

$$\mathcal{B}_i \cap \mathcal{B}_j = \{\}, \quad i \neq j, \quad i, j = 1, 2, \ldots, n_{\mathrm{bl}}. \tag{C.0.2}$$

Of course, if $j \notin \mathcal{B}_i$, it follows that $\gamma(\mathbf{s}_j, i; \omega) = 0$. Property c) implies that $|\mathcal{B}_i| \geq 2$, for all sets $\mathcal{B}_i$. Given these requirements for $\mathbf{\Gamma}(z_0)$, we define the blending noise operator

$$\mathbf{N}(z_0) \equiv (\mathbf{\Gamma}(z_0)\mathbf{\Gamma}(z_0)^{\mathrm{H}} - \mathbf{I})\mathbf{\Lambda}(z_0). \tag{C.0.3}$$

When acting on a data cube from the right side, i.e. $\mathbf{P}(z_0, z_0)\mathbf{N}(z_0)$, the blending noise operator produces a new data cube which is equal to the pseudo-deblended data with the copy of the original data removed. The result is scaled by $\mathbf{\Lambda}(z_0)$, each frequency slice of which is a diagonal matrix with elements

$$\lambda(\mathbf{s}_i, \mathbf{s}_j; \omega) = \begin{cases} \frac{1}{|\mathcal{B}_k| - 1}, & k : i \in \mathcal{B}_k,\ i = j,\ i, j = 1, 2, \ldots, n_s \\ 0 & \text{otherwise.} \end{cases} \tag{C.0.4}$$

The purpose of $\mathbf{\Lambda}(z_0)$ is to normalize for the different number of shots participating in a blended shot gather. When a shot is not blended with any other shots, the corresponding diagonal element is set to 1. The individual elements of $\mathbf{N}(z_0)$ are given by

$$n(\mathbf{s}_i, \mathbf{s}_j; \omega) = \begin{cases} \gamma(\mathbf{s}_i, k; \omega)\gamma(\mathbf{s}_j, k; \omega)^* \lambda(\mathbf{s}_j, \mathbf{s}_j; \omega), & k : i \in \mathcal{B}_k, j \in \mathcal{B}_k, i \neq j \\ (\gamma(\mathbf{s}_j, k; \omega)\gamma(\mathbf{s}_j, k; \omega)^* - 1)\lambda(\mathbf{s}_j, \mathbf{s}_j; \omega), & k : i \in \mathcal{B}_k, j \in \mathcal{B}_k, i = j \\ 0, & \text{otherwise.} \end{cases} \tag{C.0.5}$$

Since the blending code consists of pure random time delays, it holds that

$$\gamma(\mathbf{s}_j, k; \omega)\gamma(\mathbf{s}_j, k; \omega)^* - 1 = e^{j\omega\tau_{jk}} e^{-j\omega\tau_{jk}} - 1 = 0, \tag{C.0.6}$$

therefore (C.0.5) simplifies to

$$n(\mathbf{s}_i, \mathbf{s}_j; \omega) = \begin{cases} \gamma(\mathbf{s}_i, k; \omega)\gamma(\mathbf{s}_j, k; \omega)^* \lambda(\mathbf{s}_j, \mathbf{s}_j; \omega), & k : i \in \mathcal{B}_k, j \in \mathcal{B}_k \setminus i, \\ 0, & \text{otherwise.} \end{cases} \tag{C.0.7}$$

An important property of $\mathbf{N}(z_0)$ is that

$$\boldsymbol{\Gamma}(z_0) = \mathbf{N}(z_0)\boldsymbol{\Gamma}(z_0), \tag{C.0.8}$$

or, in other words, each column of $\boldsymbol{\Gamma}(z_0)$ is an eigenvector of $\mathbf{N}(z_0)$. To see why this is the case, we examine the (i, l)-th element of the product $\mathbf{N}(z_0)\boldsymbol{\Gamma}(z_0)$ which is given by

$$\sum_j n(\mathbf{s}_i, \mathbf{s}_j; \omega)\gamma(\mathbf{s}_j, l; \omega) = \tag{C.0.9}$$

$$\begin{cases} \gamma(\mathbf{s}_i, k; \omega) \sum_j \gamma(\mathbf{s}_j, k; \omega)^* \gamma(\mathbf{s}_j, l; \omega)\lambda(\mathbf{s}_j, \mathbf{s}_j; \omega), & k : i \in \mathcal{B}_k, \\ & j \in \mathcal{B}_k \setminus i \\ 0, & \text{otherwise.} \end{cases}$$

We distinguish two cases. For $l \neq k$, it follows from (C.0.1) and (C.0.2) that

$$\sum_{j \in \mathcal{B}_k \setminus i} \gamma(\mathbf{s}_j, k; \omega)^* \gamma(\mathbf{s}_j, l; \omega)\lambda(\mathbf{s}_j, \mathbf{s}_j; \omega) = 0. \tag{C.0.10}$$

For the second case, $k = l$,

$$\sum_{j \in \mathcal{B}_k \setminus i} \gamma(\mathbf{s}_j, k; \omega)^* \gamma(\mathbf{s}_j, k; \omega)\lambda(\mathbf{s}_j, \mathbf{s}_j; \omega) =$$

$$\sum_{j \in \mathcal{B}_k \setminus i} \frac{1}{|\mathcal{B}_k| - 1} = \frac{|\mathcal{B}_k \setminus i|}{|\mathcal{B}_k| - 1} = \frac{|\mathcal{B}_k| - 1}{|\mathcal{B}_k| - 1} = 1. \tag{C.0.11}$$

Using (C.0.10) and (C.0.11), (C.0.9) becomes

$$\begin{aligned}\sum_j n(\mathbf{s}_i,\mathbf{s}_j;\omega)\gamma(\mathbf{s}_j,l;\omega) &= \begin{cases}\gamma(\mathbf{s}_i,k;\omega), & k : i \in \mathcal{B}_k,\ k=l,\\ 0, & \text{otherwise.}\end{cases}\\ &= \begin{cases}\gamma(\mathbf{s}_i,l;\omega), & l : i \in \mathcal{B}_l,\\ 0, & \text{otherwise.}\end{cases}\\ &= \gamma(\mathbf{s}_i,l;\omega). \end{aligned} \tag{C.0.12}$$

Equation (C.0.12) holds for all i and j which proves (C.0.8).

Bibliography

Abma, R., and Foster, M. S., 2020, Simultaneous source seismic acquisition: Society of Exploration Geophysicists.

Abma, R., Zhang, Q., Arogunmati, A., and Beaudoin, G., 2012, An overview of BP's marine independent simultaneous source field trials: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Abma, R., Howe, D., Foster, M., Ahmed, I., Tanis, M., Zhang, Q., Arogunmati, A., and Alexander, G., 2015, Independent simultaneous source acquisition and processing: Geophysics, **80**, no. 6, WD37–WD44.

Akerberg, P., Hampson, G., Rickett, J., Martin, H., and Cole, J., 2008, Simultaneous source separation by sparse Radon transform: 78th SEG Annual International Meeting, Expanded Abstracts, 2801–2805.

Alshuhail, A., Staal, X., and Verschuur, D., 2014, Incorporating anisotropy in joint migration inversion: 84th SEG Annual International Meeting, Expanded Abstracts, 415–419.

Anderson, C., Long, A., Ziolkowski, A., Hobbs, B., and Wright, D., 2008, Multi-transient EM technology in practice: First break, **26**, no. 3.

Andersson, F., Eggenberger, K., Manen, D.-J. V., Robertsson, J., and Amundsen, L., 2016, Seismic apparition dealiasing using directionality regularization: 86th SEG Annual International Meeting, Expanded Abstracts, 56–60.

Ayeni, G., Tang, Y., and Biondi, B., 2009, Joint preconditioned least–squares inversion of simultaneous source time--lapse seismic data sets: 79th SEG Annual International Meeting, Expanded Abstracts, 3914–3918.

Ayeni, G., Almomin, A., and Nichols, D., 2011, On the separation of simultaneous–source data by inversion: 81st SEG Annual International Meeting, Expanded Abstracts, 20–25.

Baardman, R., and van Borselen, R., 2012, Separating sources in marine simultaneous shooting acquisition – method & applications: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Beasley, C., Moore, I., Monk, D., and Hansen, L., 2012, Simultaneous sources: The inaugural full-field: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Beasley, C., Moore, I., Fletcher, R., and Castellanos, C., 2016, Simultaneous source separation using adaptive robust linear algebra: 78th EAGE Conference & Exhibition, We LHR2 07.

Beasley, C. J., 2008, Simultaneous sources: A technology whose time has come: 78th SEG Annual International Meeting, Expanded Abstracts, 2796–2800.

Bednar, J., and Watt, T., February 1984, Alpha-trimmed means and their relationship to median filters: IEEE Transactions on Acoustics, Speech, and Signal Processing, **32**, no. 1, 145–153.

Bednar, J. B., 2005, A brief history of seismic migration: Geophysics, **70**, no. 3, 3MJ–20MJ.

Ben-Hadj-Ali, H., Operto, S., and Vineux, J., 2009, Three–dimensional frequency–domain full waveform inversion with phase encoding: 79th SEG Annual International Meeting, Expanded Abstracts, 2288–2292.

Berkhout, A. J., and Blacquière, G., 2013, Effect of noise in blending and deblending: 83rd SEG Annual International Meeting, Expanded Abstracts, 94–98.

Berkhout, A. J., and Blacquière, G., 2014, Combining deblending with multi–level source deghosting: 84th SEG Annual International Meeting, Expanded Abstracts, 41–45.

Berkhout, A. J., and Verschuur, D. J., 2006, Focal transformation, an imaging concept for signal restoration and noise removal: Geophysics, **71**, no. 6, A55–A59.

Berkhout, A. J., and Verschuur, D. J., 2010, Parameterization of seismic data using gridpoint responses: 80th SEG Annual International Meeting, Expanded Abstracts, 3344–3348.

Berkhout, A. J., Verschuur, D. J., and Blacquière, G., 2009, Seismic imaging with incoherent wavefields: 79th SEG Annual International Meeting, Expanded Abstracts, 2894–2898.

Berkhout, A. J., Blacquière, G., and Verschuur, D. J., 2010, Multi--scattering illumination in blended acquisition design: 80th SEG Annual International Meeting, Expanded Abstracts, 1251–1255.

Berkhout, A. J., 1982, Seismic migration, imaging of acoustic energy by wave field extrapolation, A: Theoretical aspects: Elsevier (second edition).

Berkhout, A. J., 2008, Changing the mindset in seismic data acquisition: The Leading Edge, **27**, no. 7, 924–938.

Berkhout, A. J., 2012, Blended acquisition with dispersed source arrays: Geophysics, **77**, no. 4, A19–A23.

Berkhout, A. J., 2013, Decentralized blended acquisition: 83rd SEG Annual International Meeting, Expanded Abstracts, 7–11.

Berkhout, A. G., 2014, An outlook on the future of seismic imaging, part iii: Joint Migration Inversion: Geophysical Prospecting, **62**, no. 5, 950–971.

Billette, F., and Lambaré, G., 1998, Velocity macro-model estimation from seismic reflection data by stereotomography: Geophysical Journal International, **135**, no. 2, 671–690.

Biondi, B. L., 2006, 3D seismic imaging: Society of Exploration Geophysicists.

Blacquière, G., Berkhout, A. J., and Verschuur, D. J., 2012, Double illumination in blended acquisition: 82nd SEG Annual International Meeting, Expanded Abstracts, 11–15.

Boonyasiriwat, C., and Schuster, G. T., 2010, 3D multisource full--waveform inversion using dynamic random phase encoding: 80th SEG Annual International Meeting, Expanded Abstracts, 1044–1049.

Campman, X., Tang, Z., Jamali-Rad, H., Kuvshinov, B., Danilouchkine, M., Ji, Y., Walk, W., and Smit, D., 2017, Sparse seismic wavefield sampling: The Leading Edge, **36**, no. 8, 654–660.

Candes, E. J., and Donoho, D. L., 2000, Curvelets-a surprisingly effective nonadaptive representation for objects with edges: Curves and Surfaces, Proceedings, 105–120.

Candès, E. J., and Tao, T., 2005, Decoding by linear programming: IEEE Transactions on Information Theory, **51**, no. 12, 4203–4215.

Candès, E. J., and Tao, T., 2006, Near-optimal signal recovery from random projections: Universal encoding strategies?: IEEE transactions on information theory, **52**, no. 12, 5406–5425.

Candès, E., Demanet, L., Donoho, D., and Ying, L., 2006, Fast discrete curvelet transforms: Multiscale Modeling & Simulation, **5**, no. 3, 861–899.

Cao, J., Kontakis, A., Verschuur, D. J., Gu, H., and Li, L., 2019, Deblending using the focal transformation with an efficient greedy inversion solver: Journal of Applied Geophysics.

Castellanos, C., Fletcher, R., Moore, I., and Beasley, C., 2016, Removing simultaneous source crosstalk with reverse time migration: 86th SEG Annual International Meeting, Expanded Abstracts, 4305–4309.

Červený, V., and Hron, F., 1980, The ray series method and dynamic ray tracing system for three-dimensional inhomogeneous media: Bulletin of the Seismological Society of America, **70**, no. 1, 47–77.

Červený, V., Popov, M. M., and Pšenčík, I., 1982, Computation of wave fields in inhomogeneous media – Gaussian beam approach: Geophysical Journal of the Royal Astronomical Society, **70**, no. 1, 109–128.

Chauris, H., and Nguyen, T., 2008, Seismic demigration/migration in the curvelet domain: Geophysics, **73**, no. 2, S35–S46.

Chen, S. S., and Donoho, D. L., 1998, Application of basis pursuit in spectrum estimation: IEEE International Conference on Acoustics, Speech and Signal Processing, Proceedings, 1865–1868.

Chen, S. S., Donoho, D. L., and Saunders, M. A., 2001, Atomic decomposition by basis pursuit: SIAM review, **43**, no. 1, 129–159.

Chen, Y., Caramanis, C., and Mannor, S., 2013, Robust sparse regression under adversarial corruption: 30th International Conference on Machine Learning, Proceedings, 774–782.

Chen, Y., Fomel, S., and Hu, J., 2014, Iterative deblending of simultaneous-source seismic data using seislet-domain shaping regularization: Geophysics, **79**, no. 5, V179–V189.

Chen, Y., Jin, Z., Gan, S., Yang, W., and Xiang, K., 2015, Deblending using a combined PNMO-MF-FK coherency filter: 85th SEG Annual International Meeting, Expanded Abstracts, 3920–3925.

Chen, Y., Zhang, D., Huang, W., Zu, S., Jin, Z., and Chen, W., 2016, Damped rank-reduction method for simultaneous denoising and reconstruction of 5D seismic data: 86th SEG Annual International Meeting, Expanded Abstracts, 4075–4080.

Chen, Y., Chen, H., Xiang, K., and Chen, X., 2017, Preserving the discontinuities in least-squares reverse time migration of simultaneous-source data: Geophysics, **82**, no. 3, S185–S196.

Chen, Y., 2014, Deblending using a space-varying median filter: 84th SEG Annual International Meeting, Expanded Abstracts, 82–87.

Chen, Y., 2015, Deblending by iterative orthogonalization and seislet thresholding: 85th SEG Annual International Meeting, Expanded Abstracts, 53–58.

Cheng, J., and Sacchi, M. D., 2013, Separation of simultaneous source data via iterative rank reduction: 83rd SEG Annual International Meeting, Expanded Abstracts, 88–93.

Cheng, J., and Sacchi, M. D., 2015, Separation and reconstruction of simultaneous source data via iterative rank reduction: Geophysics, **80**, no. 4, V57–V66.

Cheng, J., and Sacchi, M. D., 2016, Fast dual-domain reduced-rank algorithm for 3D deblending via randomized QR decomposition: Geophysics, **81**, no. 1, V89–V101.

Cohen, A., Dahmen, W., and DeVore, R., 2009, Compressed sensing and best k-term approximation: Journal of the American mathematical society, **22**, no. 1, 211–231.

Crosby, A., Davies, D., Fu, K., Johnston, R., Tyler, G., and Haacke, R., 2022, Smaller marine sources for higher density data: A 3D field trial in the central north sea: 83rd EAGE Conference & Exhibition, 1–5.

da Silva, K. C., Bulcão, A., Landau, L., and da Silva, J. J., 2012, A comparison of imaging conditions using de concept of blended seismic data: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Dai, W., Wang, X., and Schuster, G. T., 2011, Least-squares migration of multisource data with a deblurring filter: Geophysics, **76**, no. 5, R135–R146.

Davydenko, M., and Verschuur, D. J., 2018, Including and using internal multiples in closed-loop imaging – Field data examples: Geophysics, **83**, no. 4, R297–R305.

de Bruin, C. G. M., Wapenaar, C. P. A., and Berkhout, A. J., 1990, Angle--dependent reflectivity by means of prestack migration: Geophysics, **55**, no. 9, 1223–1234.

De Kok, R. J., and Gillespie, D., 2002, A universal simultaneous shooting technique: 64th EAGE Conference & Exhibition, Extended Abstracts, A04.

Dean, T., 2012, Establishing the limits of vibrator performance - experiments with pseudorandom sweeps: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Dellinger, J., Ross, A., Meaux, D., Brenders, A., Gesoff, G., Etgen, J., Naranjo, J., Openshaw, G., and Harper, M., 2016, Wolfspar$^{®}$, an “FWI-friendly” ultralow-frequency marine seismic source: 86th SEG Annual International Meeting, Expanded Abstracts, 4891–4895.

Donno, D., Chauris, H., and Noble, M., 2010, Curvelet-based multiple prediction: Geophysics, **75**, no. 6, WB255–WB263.

Donoho, D. L., and Elad, M., 2003, Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization: Proceedings of the National Academy of Sciences, **100**, no. 5, 2197–2202.

Donoho, D. L., 2006, For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution: Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, **59**, no. 6, 797–829.

Doulgeris, P., Mahdad, A., and Blacquière, G., 2010, Separation of blended data by iterative estimation and subtraction of interference noise: 80th SEG Annual International Meeting, Expanded Abstracts, 3514–3518.

Doulgeris, P., Blacquière, G., and Verschuur, E., 2012a, Integration of deblending and surface-related multiple elimination: application to marine data: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–6.

——— 2012b, Iterative separation of blended marine data: Discussion on the coherence--pass filter: 82nd SEG Annual International Meeting, Expanded Abstracts, 26–31.

Douma, H., and de Hoop, M. V., 2007, Leading-order seismic imaging using curvelets: Geophysics, **72**, no. 6, S231–S248.

Fabien-Ouellet, G., and Sarkar, R., 2020, Seismic velocity estimation: A deep recurrent neural-network approach: Geophysics, **85**, no. 1, U21–U29.

Farra, V., and Madariaga, R., 1988, Non-linear reflection tomography: Geophysical Journal International, **95**, no. 1, 135–147.

Ferner, R., and Sacchi, M. D., 2014, Attenuating crosstalk in blended reverse time migration via m-estimators: 84th SEG Annual International Meeting, Expanded Abstracts, 4038–4042.

Fomel, S., 2002, Applications of plane–wave destruction filters: Geophysics, **67**, no. 6, 1946–1960.

Foucart, S., and Rauhut, H., 2017, A mathematical introduction to compressive sensing: Springer.

Fred AminzadehJean Brac, T. K., 1997, 3-d salt and overthrust models: Society of Exploration Geophysicists.

Gan, S., Wang, S., Chen, X., and Chen, Y., 2015, Deblending using a structural-oriented median filter: 85th SEG Annual International Meeting, Expanded Abstracts, 59–64.

Gao, F., Atle, A., and Williamson, P., 2010, Full waveform inversion using deterministic source encoding: 80th SEG Annual International Meeting, Expanded Abstracts, 1013–1017.

Garotta, R., 1983, Simultaneous recording of several Vibroseis® seismic lines: 53rd SEG Annual International Meeting, Expanded Abstracts, 308–310.

Gauthier, O., Virieux, J., and Tarantola, A., 1986, Two--dimensional nonlinear inversion of seismic waveforms: Numerical results: Geophysics, **51**, no. 7, 1387–1403.

Gazdag, J., and Sguazzero, P., 1984, Migration of seismic data by phase shift plus interpolation: Geophysics, **49**, no. 2, 124–131.

Gisolf, D., and Verschuur, E., 2010, The principles of quantitative acoustical imaging: EAGE Publications BV.

Godwin, J., and Sava, P., 2010, Blended source imaging by amplitude encoding: 80th SEG Annual International Meeting, Expanded Abstracts, 3125–3129.

Godwin, J., and Sava, P., 2011, A comparison of shot--encoding schemes for wave--equation migration: 81st SEG Annual International Meeting, Expanded Abstracts, 32–36.

Gribonval, R., Rauhut, H., Schnass, K., and Vandergheynst, P., 2008, Atoms of all channels, unite! average case analysis of multi-channel sparse recovery using greedy algorithms: Journal of Fourier analysis and Applications, **14**, no. 5-6, 655–687.

Gulati, J. S., Salama, A., Leaney, S. W., Beasley, C. J., Coste, E., Menkiti, H., and Tulett, J., 2011, Faster 3D VSP acquisition using simultaneous sources: 81st SEG Annual International Meeting, Expanded Abstracts, 4249–4252.

Hampson, G., Stefani, J., and Herkenhoff, F., 2008, Acquisition using simultaneous sources: 78th SEG Annual International Meeting, Expanded Abstracts, 2816–2820.

He, Q., and Wang, Y., 2021, Reparameterized full-waveform inversion using deep neural networks: Geophysics, **86**, no. 1, V1–V13.

Henin, G., Marin, D., Maitra, S., Rollet, A., Chandola, S. K., Kumar, S., Kady, N. E., and Foo, L. C., 2015, Deblending 4-component simultaneous-source data – a 2D OBC case study in Malaysia: 85th SEG Annual International Meeting, Expanded Abstracts, 43–47.

Hennenfent, G., Fenelon, L., and Herrmann, F. J., 2010, Nonequispaced curvelet transform for seismic data reconstruction: A sparsity-promoting approach: Geophysics, **75**, no. 6, WB203–WB210.

Herrmann, F. J., Wang, D., and Verschuur, D. J. E., 2008, Adaptive curvelet-domain primary-multiple separation: Geophysics, **73**, no. 3, A17–A21.

Herrmann, F. J., Erlangga, Y. A., and Lin, T. T., 2009, Compressive simultaneous full-waveform simulation: Geophysics, **74**, no. 4, A35–A40.

Howe, D., Foster, M., Allen, T., Taylor, B., and Jack, I., 2008, Independent simultaneous sweeping-a method to increase the productivity of land seismic crews: 78th SEG Annual International Meeting, Expanded Abstracts, 2826–2830.

Hu, J., Wang, X., Wang, H., Tian, L., and Zhou, D., 2011, Frequency grouping coding scheme for blended source imaging: 81st SEG Annual International Meeting, Expanded Abstracts, 3404–3408.

Huo, S., Luo, Y., and Kelamis, P. G., 2012, Simultaneous sources separation via multidirectional vector–median filtering: Geophysics, **77**, no. 4, V123–V131.

Ibrahim, A., and Sacchi, M. D., 2013, Simultaneous source separation using a robust Radon transform: Geophysics, **79**, no. 1, V1–V11.

Ibrahim, A., and Sacchi, M. D., 2015, Fast simultaneous seismic source separation using Stolt migration and demigration operators: Geophysics, **80**, no. 6, WD27–WD36.

Jeong, W., Tsingas, C., and Almubarak, M. S., 2022, Deblending and merging of 3d multi-sweep seismic blended data: Geophysical Prospecting, **70**, no. 2, 246–261.

Ji, Y., Kragh, E., and Christie, P., 2012, A new simultaneous source separation algorithm using frequency-diverse filtering: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Jiang, Z., and Abma, R., 2010, An analysis on the simultaneous imaging of simultaneous source data: 80th SEG Annual International Meeting, Expanded Abstracts, 3115–3119.

Kabir, M. M. N., and Verschuur, D. J., 1995, Restoration of missing offsets by parabolic Radon transform: Geophysical Prospecting, **43**, no. 3, 347–368.

Kelly, K. R., Ward, R. W., Treitel, S., and Alford, R. M., 1976, Synthetic seismograms: A finite--difference approach: Geophysics, **41**, no. 1, 2–27.

Kinneging, N. A., Budejicky, V., Wapenaar, C. P. A., and Berkhout, A. J., 1989, Efficient 2D and 3D shot record redatuming: Geophysical Prospecting, **37**, no. 5, 493–530.

Kontakis, A., and Verschuur, D. J., 2014, Deblending via sparsity-constrained inversion in the focal domain: 76th EAGE Conference & Exhibition, Extended Abstracts, Th ELI2 02.

Kontakis, A., and Verschuur, D. J., 2015, Deblending via a hybrid focal and linear Radon transform: 77th EAGE Conference & Exhibition, Extended Abstracts, We N101 02.

Kontakis, A., and Verschuur, D. J., 2016, Focal deblending using smart subsets of towed streamer 5D data: 86th SEG Annual International Meeting, Expanded Abstracts, 4612–4617.

Kontakis, A., and Verschuur, D. J., 2017a, Focal deblending using smart subsets of OBN 5D data: 79th EAGE Conference & Exhibition, We P3 11.

——— 2017b, Using a hybrid focal – curvelet transform for deblending: 87th SEG Annual International Meeting, Expanded Abstracts, 4903–4908.

Kontakis, A., Wu, S., and Verschuur, D. J., 2016, Acquisition geometry-aware focal deblending: 78th EAGE Conference & Exhibition, Th P4 05.

Krebs, J. R., Anderson, J. E., Hinkley, D., Neelamani, R., Lee, S., Baumstein, A., and Lacasse, M.-D., 2009, Fast full-wavefield seismic inversion using encoded sources: Geophysics, **74**, no. 6, WCC177–WCC188.

Krupovnickas, T., Matson, K., Corcoran, C., and Pascual, R., 2012, Marine simultaneous source OBS survey suitability for 4D analysis: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Kumar, R., Wason, H., and Herrmann, F. J., 2015, Source separation for simultaneous towed-streamer marine acquisition — A compressed sensing approach: Geophysics, **80**, no. 6, WD73–WD88.

Kumar, R., Sharan, S., Wason, H., and Herrmann, F., 2016, Time-jittered marine acquisition: A rank-minimization approach for 5D source separation: 86th SEG Annual International Meeting, Expanded Abstracts, 119–123.

Kutscha, H., and Verschuur, D. J., 2012, Data reconstruction via sparse double focal transformation: An overview: Signal Processing Magazine, IEEE, **29**, no. 4, 53–60.

Kutscha, H., 2014, The double focal transformation and its application to data reconstruction: Ph.D. thesis, Delft University of Technology.

Landrum, R. A., 1987, Simultaneous performance of multiple seismic vibratory surveys: US Patent 4 715 020.

Leader, C., and Biondi, B., 2014, Demigration and image space separation of simultaneously acquired data: 84th SEG Annual International Meeting, Expanded Abstracts, 88–93.

Lee, S., Hinkley, D., Krebs, J. R., and Anderson, J. E., 2012, Crosstalk noise analysis of simultaneous-source full wavefield inversion: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Lempitsky, V., Vedaldi, A., and Ulyanov, D., 2018, Deep image prior: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9446–9454.

Li, W., and Zhao, Y., 2015, An imaging perspective of low-rank seismic data interpolation and denoising: 85th SEG Annual International Meeting, Expanded Abstracts, 4106–4110.

Li, X., Aravkin, A. Y., van Leeuwen, T., and Herrmann, F. J., 2012, Fast randomized full-waveform inversion with compressive sensing: Geophysics, **77**, no. 3, A13–A17.

Li, C., Huang, J., and Li, Z., 2016, Regularized least-squares reverse time migration of simultaneous-source data with adaptive singular spectrum analysis: 86th SEG Annual International Meeting, Expanded Abstracts, 4419–4423.

Li, C., Mosher, C. C., and Ji, Y., 2019, An amplitude-preserving deblending approach for simultaneous sources: Geophysics, **84**, no. 3, V185–V196.

Li, Z., 2020, Adaptive multiple subtraction based on support vector regression: Geophysics, **85**, no. 1, V57–V69.

Liu, Y., Luo, Y., and Wang, Y., 2009, Vector median filter and its applications in geophysics: 79th SEG Annual International Meeting, Expanded Abstracts, 3342–3346.

Liu, Z., Wang, B., Specht, J., Sposato, J., and Zhai, Y., 2014, Enhanced adaptive subtraction method for simultaneous source separation: 84th SEG Annual International Meeting, Expanded Abstracts, 115–119.

Long, A., von Abendorff, E., Purves, M., Norris, J., and Moritz, A., 2013, Simultaneous long offset (SLO) towed streamer seismic acquisition: 23rd ASEG Geophysical Conference, Extended Abstracts, 1–4.

Luiken, N., Ravasi, M., and Birnie, C. E., 2022, A hybrid approach to seismic deblending: when physics meets self-supervision: arXiv preprint arXiv:2205.15395.

Lumley, D. E., 2001, Time-lapse seismic reservoir monitoring: Geophysics, **66**, no. 1, 50–53.

Mahdad, A., Doulgeris, P., and Blacquière, G., 2011, Separation of blended data by iterative estimation and subtraction of blending interference noise: Geophysics, **76**, no. 3, Q9–Q17.

Makhorin, M., Fontana, P., Podshuveyt, N., Peter, D. S., Williams, T., and Smit, F., 2013, Dual streamer vessel simultaneous 3D acquisition in Greenland: 83rd SEG Annual International Meeting, Expanded Abstracts, 206–210.

Maraschini, M., Dyer, R., Stevens, K., Bird, D., and King, S., 2012, An iterative SVD method for deblending: theory and examples: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Mavko, G., Mukerji, T., and Dvorkin, J., 2009, The rock physics handbook: Tools for seismic analysis of porous media: Cambridge university press.

Meunier, J., 2011, Seismic acquisition from yesterday to tomorrow: Society of Exploration Geophysicists.

Michelot, C., 1986, A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$: Journal of Optimization Theory and Applications, **50**, no. 1, 195–200.

Moldoveanu, N., Ji, Y., and Beasley, C., 2012, Multivessel coil shooting acquisition with simultaneous sources: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–6.

Moore, I., Dragoset, B., Ommundsen, T., Wilson, D., Ward, C., and Eke, D., 2008, Simultaneous source separation using dithered sources: 78th SEG Annual International Meeting, Expanded Abstracts, 2806–2810.

Moore, I., Bisley, R., Beasley, C., Bayly, M., Monk, D., and Hansen, L., 2013, Simultaneous-source acquisition and environmental noise - good or bad?: 75th EAGE Conference & Exhibition, Extended Abstracts, Th 08 03.

Moore, I., Fletcher, R., Beasley, C., and Castellanos, C., 2016, Data studies of simultaneous source separation using robust linear algebra: 86th SEG Annual International Meeting, Expanded Abstracts, 4623–4627.

Moore, I., Beasley, C., and Fletcher, R., 2017, The roles of sparseness and robust linear algebra in simultaneous-source separation: 79th EAGE Conference & Exhibition, We P3 10.

Mueller, M. B., Halliday, D. F., van Manen, D.-J., and Robertsson, J. O. A., 2015, The benefit of encoded source sequences for simultaneous source separation: Geophysics, **80**, no. 5, V133–V143.

Mueller, M. B., Halliday, D. F., van Manen, D.-J., and Robertsson, J. O. A., 2016, Optimizing near-orthogonal air-gun firing sequences for marine simultaneous source separation: Geophysics, **81**, no. 6, V415–V423.

Naghizadeh, M., and Sacchi, M., 2018, Ground-roll attenuation using curvelet downscaling: Geophysics, **83**, no. 3, V185–V195.

Neelamani, R. N., Krohn, C. E., Krebs, J. R., Romberg, J. K., Deffenbaugh, M., and Anderson, J. E., 2010, Efficient seismic forward modeling using simultaneous random sources and sparsity: Geophysics, **75**, no. 6, WB15–WB27.

Operto, S., Virieux, J., Amestoy, P., L'Excellent, J.-Y., Giraud, L., and Ali, H. B. H., 2007, 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study: Geophysics, **72**, no. 5, SM195–SM211.

Paramo, P., Vincent, K., Cegna, A., Kommedal, J., Napier, P., and Cardinez, S., 2013, AVO analysis of independent simultaneous source OBC data from Trinidad: 83rd SEG Annual International Meeting, Expanded Abstracts, 368–372.

Pawlowski, B., 1998, Gravity gradiometry in resource exploration: The Leading Edge, **17**, no. 1, 51–52.

Peng, C., and Meng, J., 2016, Inversion-based 3D deblending of towed-streamer simultaneous source data using sparse tau-p and wavelet transforms: 86th SEG Annual International Meeting, Expanded Abstracts, 4607–4611.

Peng, C., Liu, B., Khalil, A., and Poole, G., 2013, Deblending of simulated simultaneous sources using an iterative approach: An experiment with variable-depth streamer data: 83rd SEG Annual International Meeting, Expanded Abstracts, 4278–4282.

Poole, G., Russier, D., Stevens, K., Mensch, T., and Siliqi, R., 2014, A deblending strategy using alternating constant delay simultaneous source data: 84th SEG Annual International Meeting, Expanded Abstracts, 4335–4339.

Pramik, B., Bell, M. L., Grier, A., and Lindsay, A., 2015, Field testing the AquaVib: an alternate marine seismic source: 85th SEG Annual International Meeting, Expanded Abstracts, 181–185.

Pritchett, W. C., 1991, An example of simultaneous recording where necessary signal separation is easily achieved: Geophysics, **56**, no. 1, 9–17.

Rao, R. M., and Dianat, S. A., 2005, Basics of code division multiple access (CDMA): SPIE Press.

Robein, E., 2010, Seismic imaging: A review of the techniques, their principles, merits and limitations: European Association of Geoscientists and Engineers.

Robertsson, J. O. A., Amundsen, L., and Pedersen, Å. S., 2016a, Signal apparition for simultaneous source wavefield separation: Geophysical Journal International, **206**, no. 2, 1301–1305.

——— 2016b, Wavefield signal apparition: Simultaneous source separation: 86th SEG Annual International Meeting, Expanded Abstracts, 102–106.

Sallas, J. J., Corrigan, D., and Allen, K. P., 1998, High fidelity vibratory source seismic method with source separation: US Patent 5 721 710.

Sallas, J. J., Gibson, J. B., Lin, F., Winter, O., Montgomery, B., and Nagarajappa, P., 2008, Broadband vibroseis using simultaneous pseudorandom sweeps: 78th SEG Annual International Meeting, Expanded Abstracts, 100–104.

Sardy, S., Tseng, P., and Bruce, A., 2001, Robust wavelet denoising: IEEE Transactions on Signal Processing, **49**, no. 6, 1146–1152.

Schleicher, J., Novais, A., and Costa, J. C., 2013, Reduction of crosstalk in blended-shot migration: 83rd SEG Annual International Meeting, Expanded Abstracts, 3937–3941.

Schmidt, L., Indyk, P., Chen, C., Cyr, A. S., and Hohl, D., 2013, Shot encoding with random projections: 83rd SEG Annual International Meeting, Expanded Abstracts, 3979–3983.

Schostak, B., and Jenkerson, M., 2015, The marine vibrator joint industry project: 85th SEG Annual International Meeting, Expanded Abstracts, 4961–4962.

Schuster, G. T., Dai, W., Zhan, G., and Boonyasiriwat, C., 2010, Theory of multisource crosstalk reduction by phase-encoded statics: 80th SEG Annual International Meeting, Expanded Abstracts, 3110–3114.

Sen, S., Liu, Z., Sheng, J., and Wang, B., 2014, Reweighted thresholding and orthogonal projections for simultaneous source separation: 84th SEG Annual International Meeting, Expanded Abstracts, 243–248.

Shipilova, E., Barone, I., Hugonnet, P., Dupinet, C., Boelle, J.-L., Giboli, M., and Piazza, J.-L., 2016, Simultaneous-source seismic acquisitions: Do they allow reservoir characterization? A feasibility study with blended onshore real data: 86th SEG Annual International Meeting, Expanded Abstracts, 107–112.

Silverman, D., 1979, Method of three dimensional seismic prospecting: US Patent 4 159 463.

Son, W., Pyun, S., Jeong, W., and Min, D.-J., 2012, Simultaneous source full waveform inversion using robust objective function: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–6.

Spitz, S., Hampson, G., and Pica, A., 2008, Simultaneous source separation: A prediction--subtraction approach: 78th SEG Annual International Meeting, Expanded Abstracts, 2811–2815.

Stork, C., and Clayton, R., 1991, Linear aspects of tomographic velocity analysis: Geophysics, **56**, no. 4, 483–495.

Sun, Y., and Verschuur, E., 2018, Three-dimensional receiver deghosting of seismic streamer data using l1 inversion and redundant extended radon dictionary: Geophysical Prospecting, **66**, no. 5, 987–1003.

Sun, J., Slang, S., Elboth, T., Larsen Greiner, T., McDonald, S., and Gelius, L.-J., 2020, A convolutional neural network approach to deblending seismic data: Geophysics, **85**, no. 4, WA13–WA26.

Sun, J., Hou, S., Vinje, V., Poole, G., and Gelius, L.-J., 2022, Deep learning-based shot-domain seismic deblending: Geophysics, **87**, no. 3, V215–V226.

Tang, Y., and Biondi, B., 2009, Least--squares migration/inversion of blended data: 79th SEG Annual International Meeting, Expanded Abstracts, 2859–2863.

Telford, W. M., Geldart, L. P., and Sheriff, R. E., 1990, Applied geophysics: Cambridge university press.

Thorbecke, J. W., Wapenaar, C., and Swinnen, G., 2004, Design of one-way wavefield extrapolation operators, using smooth functions in WLSQ optimization: Geophysics, **69**, no. 4, 1037–1045.

Toushmalani, R., and Saibi, H., 2015, 3D gravity inversion using Tikhonov regularization: Acta Geophysica, **63**, no. 4, 1044–1065.

Trad, D., Siliqi, R., Poole, G., and Boelle, J.-L., 2012, Fast and robust deblending using apex shifted radon transform: 82nd SEG Annual International Meeting, Expanded Abstracts, 1–5.

Tropp, J., 2004, Greed is good: algorithmic results for sparse approximation: IEEE Transactions on Information Theory, **50**, no. 10, 2231–2242.

Tsingas, C., Kim, Y. S., and Yoo, J., 2016, Broadband acquisition, deblending, and imaging employing dispersed source arrays: The Leading Edge, **35**, no. 4, 354–360.

Tsingas, C., Almubarak, M. S., Jeong, W., Al Shuhail, A., and Trzesniowski, Z., 2020, 3D distributed and dispersed source array acquisition and data processing: The leading edge, **39**, no. 6, 392–400.

Vamaraju, J., and Sen, M. K., 2019, Unsupervised physics-based neural networks for seismic migration: Interpretation, **7**, no. 3, SE189–SE200.

van den Berg, E., and Friedlander, M. P. SPGL1: A solver for large-scale sparse reconstruction:, June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

van den Berg, E., and Friedlander, M. P., 2008, Probing the pareto frontier for basis pursuit solutions: SIAM Journal on Scientific Computing, **31**, no. 2, 890–912.

van den Berg, E., and Friedlander, M. P., October 2011, Sparse optimization with least-squares constraints: SIAM Journal on Optimization, **21**, no. 4, 1201–1229.

van Mastrigt, P., Vaage, S., Dunn, M., and Pramik, B., 2002, Improvements in 3-D marine acquisition using continuous long offset (CLO): The Leading Edge, **21**, no. 4, 394–399.

Verschuur, D. J., and Berkhout, A. J., 2009, Target–oriented, least–squares imaging of blended data: 79th SEG Annual International Meeting, Expanded Abstracts, 2889–2893.

Verschuur, D., and Berkhout, A., 2011, Seismic migration of blended shot records with surface-related multiple scattering: Geophysics, **76**, no. 1, A7–A13.

Verschuur, D. J., Berkhout, A., and Wapenaar, C., 1992, Adaptive surface-related multiple elimination: Geophysics, **57**, no. 9, 1166–1177.

Virieux, J., and Operto, S., 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, no. 6, WCC1–WCC26.

Walker, C. D. T., Hays, D., and Monk, D., 2013, Blended source single sided full azimuth ocean bottom seismic acquisition: 83rd SEG Annual International Meeting, Expanded Abstracts, 157–160.

Walker, C., Ajlani, G., Hall, M., Al Masaabi, S., Al Kobaisi, A., Casson, G., and Hagiwara, H., 2017, Application of blended sources offshore Abu Dhabi: First Break, **35**, no. 11.

Wang, B., Li, J., and Han, D., 2022a, Iterative deblending using MultiResUNet with multilevel blending noise for training and transfer learning: Geophysics, **87**, no. 3, V205–V214.

——— 2022b, A self-supervised deep learning method for seismic data deblending using a blind-trace network: IEEE Transactions on Neural Networks and Learning Systems, pages 1–10.

Wapenaar, K., van der Neut, J., and Thorbecke, J., 2012, Deblending by direct inversion: Geophysics, **77**, no. 3, A9–A12.

Wapenaar, C. P. A., 1984, Elastic wave field extrapolation: Redatuming of single- and multi-component seismic data: Elsevier.

Wason, H., and Herrmann, F. J., 2013, Time-jittered ocean bottom seismic acquisition: 83rd SEG Annual International Meeting, Expanded Abstracts, 1–6.

Wason, H., Kumar, R., Herrmann, F. J., and Aravkin, A. Y., 2014, Source separation via SVD-free rank minimization in the hierarchical semi-separable representation: 84th SEG Annual International Meeting, Expanded Abstracts, 120–126.

Wei, J., 2013, Comparing the MEMS accelerometer and the analog geophone: The Leading Edge, **32**, no. 10, 1206–1210.

Whitmore, N. D., Valenciano, A. A., Sollner, W., and Lu, S., 2010, Imaging of primaries and multiples using a dual-sensor towed streamer: 80th SEG Annual International Meeting, Expanded Abstracts, 3187–3192.

Wong, J., and Langton, D., 2014, Field testing of multiple simultaneous vibrator sources controlled by m-sequences: 84th SEG Annual International Meeting, Expanded Abstracts, 173–177.

Wong, J., and Langton, D., 2015, Field and numerical investigation of filtered m-sequence pilots for vibroseis acquisition: 85th SEG Annual International Meeting, Expanded Abstracts, 196–200.

Wong, J., 2013, Multiple simultaneous vibrators controlled by m-sequences: 83rd SEG Annual International Meeting, Expanded Abstracts, 109–113.

Wu, S., Blacquière, G., and van Groenestijn, G.-J., 2015, Shot repetition: an alternative approach to blending in marine seismic: 85th SEG Annual International Meeting, Expanded Abstracts, 120–126.

Wu, S., Groenestijn, G.-J. V., and Blacquière, G., 2016, Using the source ghost in blended marine acquisition: 86th SEG Annual International Meeting, Expanded Abstracts, 87–91.

Xu, W., Lipari, V., Bestagini, P., Ravasi, M., Chen, W., and Tubaro, S., 2022, Intelligent seismic deblending through deep preconditioner: IEEE Geoscience and Remote Sensing Letters, **19**, 1–5.

Yang, F., and Ma, J., 2019, Deep-learning inversion: A next-generation seismic velocity model building method: Geophysics, **84**, no. 4, R583–R599.

Yu, Z., Abma, R., Etgen, J., and Sullivan, C., 2017, Attenuation of noise and simultaneous source interference using wavelet denoising: Geophysics, **82**, no. 3, V179–V190.

Yu, S., Ma, J., and Wang, W., 2019, Deep learning for denoising: Geophysics, **84**, no. 6, V333–V350.

Yuan, S., Liu, J., Wang, S., Wang, T., and Shi, P., 2018, Seismic waveform classification and first-break picking using convolution neural networks: IEEE Geoscience and Remote Sensing Letters, **15**, no. 2, 272–276.

Zhan, C., Malik, R., Specht, J., Liu, Z., and Teixeira, D., 2015, Deblending of continuously recorded OBN data by subtraction integrated with a median filter: 85th SEG Annual International Meeting, Expanded Abstracts, 4673–4677.

Zhang, J., and Toksöz, M. N., 1998, Nonlinear refraction traveltime tomography: Geophysics, **63**, no. 5, 1726–1737.

Zhang, Q., Abma, R., and Ahmed, I., 2013a, A marine node simultaneous source acquisition trial at Atlantis: 83rd SEG Annual International Meeting, Expanded Abstracts, 99–103.

——— 2013b, Separation of ISS seismic data via vector median filter in T-X and F-X domains: 83rd SEG Annual International Meeting, Expanded Abstracts, 4377–4381.

Zhang, Z., Xu, Y., Yang, J., Li, X., and Zhang, D., 2015, A survey of sparse representation: algorithms and applications: IEEE access, **3**, 490–530.

Zhou, Y., Chen, W., Gao, J., and Pascal, F., 2013, Seismic deblending by sparse inversion over dictionary learning: 83rd SEG Annual International Meeting, Expanded Abstracts, 273–278.

Zu, S., Zhou, H., Chen, Y., Chen, H., Cao, M., and Xie, C., 2016a, A marine field trial for iterative deblending of simultaneous sources: 86th SEG Annual International Meeting, Expanded Abstracts, 113–118.

——— 2016b, A periodically varying code for improving deblending of simultaneous sources in marine acquisition: Geophysics, **81**, no. 3, V213–V225.

Zu, S., Zhou, H., Li, Q., Chen, H., Zhang, Q., Mao, W., and Chen, Y., 2017, Shot-domain deblending using least-squares inversion: Geophysics, **82**, no. 4, V241–V256.

Zu, S., Cao, J., Qu, S., and Chen, Y., 2020, Iterative deblending for simultaneous source data using the deep neural network: Geophysics, **85**, no. 2, V131–V141.

Acknowledgments

In this final part of the thesis I would like to express my gratitude to the many people whose support over the years enabled me to complete it. Pivotal to my decision to pursue a Ph.D. was the encouragement I received from Xander Campman, Zijian Tang and Mikhail Danilouchkine from Shell Global Solutions. Xander, Zijian and Mikhail, thank you both for this and also for contributing ideas, suggestions and corrections that improved my master's thesis.

One of the first memories I have from the applied sciences department was of myself knocking on a green office door and entering the office of Eric Verschuur, my then soon-to-be daily supervisor. When I entered the office, I still had some doubts as to whether I should continue with graduate studies. I left the office entirely convinced, with an exciting research question and looking forward to my first day as a researcher. Eric, thank you for being a wonderful supervisor. Your ability to debug implementation issues by subtle hints in the results is uncanny. Your office door was always open for us students and if I regret one thing, it is that I did not cross it often enough, stubbornly spending days trying to solve issues that would have only taken a few minutes to resolve had I just asked you. As it happens to many, when the time came for me to write my thesis, the task appeared insurmountable and very intimidating. It was only through the biweekly meetings that you kindly agreed to that I was able to complete the manuscript at the end. I tend to always see the glass half-empty, often due to my lack of experience. You showed me that that things were sometimes better than I thought, by putting things into perspective. Presenting one's work in front of an audience can be a very stressful task for students especially in the beginning. I would also like to thank you for your encouragement and feedback after presentations that helped me both improve and be more relaxed when presenting.

Three or four days after I joined the team, there was the first Delphi sponsor meeting I attended as a spectator. It was the first time I was introduced to Gerrit Blacquière. Gerrit gave me a piece of advice that I still religiously follow to this day. Whenever I have to give a presentation, I always visit the presentation room earlier and get acquainted with the space and how it looks when standing on the podium. It has helped me immensely and it is an advice I always give to others too. Gerrit thank you for being supportive and for all the interesting discussions we had during the sponsor meetings. On the same day I also met Prof. Augustinus J. Berkhout, who presented me with the challenge of completing my Ph.D. in less time than it took him to complete his. On that front I am afraid I have failed spectacularly. Dear Prof. Berkhout, I will always remember and admire your strategic, bird's eye view on things. Prof. Dries Gisolf, we had met a few months prior to my starting date, as you were a member of my master's thesis defense committee. I endearingly refer to the book you coauthored with Eric as the 'green book'. It served as my first introduction to seismic migration and related concepts and as such, has earned a special place in my heart. I still consult it to this day.

Our work as researchers requires that certain infrastructure is in place and is well-functioning. Thank you Edo Bergsma, Henry den Bok and Ronald Ligteringen for making sure we had everything we needed and for dealing very promptly with technical issues that would occasionally arise. I would also like to thank Margaret van Fessem, Angela van der Sande and Annelies van Beek for helping us students out with the university-side of things.

Discovering the strengths and weaknesses of new seismic processing methods would not be possible without having some real world data to test against. CGG, TGS and Equinor kindly provided the field data used in the examples present in this thesis. I extend my thanks to them and the sponsors of the Delphi consortium for their generous support. Dear committee members prof. Evert Slob, Eric Verschuur, prof. Geert Leus, prof. Wim Mulder, prof. Jörg Schleicher, Gerrit Blacquière and Matteo Ravasi: I imagine that the perfect thesis manuscript is a very rare sight. The better-than-it-was-before manuscript, on the other hand, is almost always within the realm of possibility. Thank you for accepting to be members of the defense committee and for suggesting corrections and improvements for the manuscript.

Gabriel, Mikhail, Khalid, Aparajita, Ana and Ulaş: we shared the same office which means we had tons of discussions about all things technical and otherwise. I enjoyed every minute of it. Gabriel, Shan, Ulaş and Abdulrahman: everyday I would pass by the offices at 11:37 and would ask "are you joining for lunch". Saying this entire phrase required too much energy from my part, so over time I shortened it to a mere "lunch?". You almost always joined, which made me very happy. Abdulrahman, Shan, Gabrio, Alok, Aayush, Shogo, Xander, Nick, Farid, Halah, Bouchaib, Hussain, Amarjeet, Tomohide, Matteo, Sixue, Shotaro, Runhai, Siddarth, Özkan, Jan-Willem: fellow Delphi colleagues, discussing our research projects helped put the work I tried to do into context and allowed me to see its connections with the other types of processing that you were working on. Having met many times at sponsor meetings and conferences means we have many shared experiences, which I cherish and remember with joy. Coming from a different background, I can truly say that I learned many of the things I know about our field by interacting with you.

Manuel and Junhai, I partially supervised work that you did in our group, so we collaborated more closely. Your hard work and commitment helped also further my understanding of the strengths and limitations of various approaches to wavefield separation. For this I am indebted to you. Myrna, Joeri, Max, Carlos, Iris, Boris, Joost, Niels: although in a different group and a different building, I am glad I had the chance to spend time with you in courses and events outside the university. I was always excited whenever I heard that you were joining the same conferences. I fondly remember the SEG meeting in Dallas and also visiting Fledermaus in Vienna per Niels' suggestion. Excellent choice Niels!

Current Delphi PhD candidates and post docs Billy, Dong, Camille, Ali, Leo, Siamak, Andreas, Dieter, Andrea, Azin, Tiexing, Lele, most you joined after I was out of the office, so we did not get to interact daily. I wish you success with your research and, for those of you who have defended yet, I am looking forward to seeing you at your own graduation ceremony.

I extend many thanks to colleagues from the past and present at my current employer: Daniele, Erşan, Ernesto, Taqi, Rob, Roald, Matteo, Rolf, Marcin, Mikhail, Chris, Hannes, Jewoo, Paul, Ali, Diego and Yimin, thank you for your continued support as I was finalizing the manuscript. The daily lunch conversations we have with many of you directly influenced the propositions that accompanied this thesis. If I could say I learned half the geophysics I know by interacting with my Ph.D. cohort, the other half I certainly learned from you.

Dan and Mihai, my friends from the EE times, we lived together in Rotterdam and it seems like a lifetime ago. While doing my Ph.D. I visited Bucharest two times, once each time one of got married. What a fantastic time I had! Wilfred and Siavash you were also there, but you I had the chance to meet more often here in the Netherlands too. I hope that the meetings at Pizza Dennis for some catching up and some high-cholesterol philosophizing™ will keep happening for years to come. Ioannis and Vasso, we had quite a few get-togethers as well. I still marvel at Ioannis' hyper disciplined way of Ph.D.-ing. It was a gold standard example that I tried to emulate, but I was not up to the task as I quickly found out. Sigvald and Marianne: I once received an invitation to join you for hiking around Tafjordfjella, Norway. I will always remember this hiking trip and the kind and amazing hosts you were.

Dear Mafalda, Vlad and Alexandra, I have played many times board games with you. I almost never won, but I got to meet and spend time with you, which for me is the important win. We've also been to numerous film festivals together, I was always curious to hear your opinion about each film afterwards. You always were very welcoming and I enjoyed every second we spent together. Remko and Fatma, I also want to thank you here, for spending new year's eve with me last year, when it was not easy to go back home. I have many nice memories and I am sure I'll make some new ones too. Dear Sandra: perhaps I will one day be adopted by a cat, but I doubt it. They are too perfect. Meanwhile I have to be content with myself being only an imperfect human. Thank you for joining for lunch back when you were in

Delft and for being a such a kind friend all these years.

My numerous friends and extended family members from back home: at the time of writing this, I reckon we are spread over two continents and five different countries. This means that some of you I do not meet often, yet you have always been there for me in some way. Your love, support and encouragement and good friendship have been critical for me to keep going when the going got tough. I would like to thank you for your support and understanding from the depths of my heart.

Apostolos Kontakis