

IDEA League

MASTER OF SCIENCE IN APPLIED GEOPHYSICS
RESEARCH THESIS

Seismic deblending of simultaneous crossline sources

Christian Reinicke Urruticoechea

July 17, 2015

Seismic deblending of simultaneous crossline sources

MASTER OF SCIENCE THESIS

for the degree of Master of Science in Applied Geophysics at
Delft University of Technology
ETH Zürich
RWTH Aachen University
by

Christian Reinicke Urruticoechea

July 17, 2015

Department of Geoscience & Engineering	.	Delft University of Technology
Department of Earth Sciences	.	ETH Zürich
Faculty of Georesources and Material Engineering	.	RWTH Aachen University



Delft University of Technology

Copyright © 2013 by IDEA League Joint Master's in Applied Geophysics:

Delft University of Technology, ETH Zürich, RWTH Aachen University

All rights reserved.

No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying or by any information storage and retrieval system, without permission from this publisher.

Printed in The Netherlands, Switzerland, Germany

IDEA LEAGUE
JOINT MASTER'S IN APPLIED GEOPHYSICS

Delft University of Technology, The Netherlands
ETH Zürich, Switzerland
RWTH Aachen, Germany

Dated: *July 17, 2015*

Supervisor(s):

Dr. Ir. G.J.A. van Groenestijn

Dr. Ir. G.G.Drijkoningen

Committee Members:

Dr. Ir. G.J.A. van Groenestijn

Dr. Ir. G.G.Drijkoningen

Dr. Christoph Clauser

Abstract

This is the shortest abstract you have ever seen.

Acknowledgements

First of all I want to thank all the people who have participated in this project .. Remember, often more people have contributed to your final thesis than you initially would think of.

Delft, University of Technology
July 17, 2015

Christian Reinicke Urruticoechea

Table of Contents

Abstract	v
Acknowledgements	vii
Nomenclature	xvii
Acronyms	xvii
1 Introduction	1
2 Theory	3
2-1 The Forward Model of Blending	3
2-1-1 Conventional Seismic Data	3
2-1-2 Blended Seismic Data	5
2-2 Deblending	6
2-2-1 Pseudo-Deblending	6
2-2-2 Common Receiver Gather	7
2-2-3 Iterative Estimation of Blending Noise	7
3 Incoherency	13
3-1 Analysis of the Blending Matrix	13
3-2 Results Spatial and Temporal Incoherency	16
3-3 Effect of Incoherency	17
3-4 Results Incoherency	19
3-5 Effect of Maximum Firing Time Delay	20
3-6 Results Maximum Firing Time Delay	20
3-7 Conclusions	21

4 Crossline Deblending (3D)	23
4-1 Data Sorting	23
4-2 3D f-k-k Filter	26
4-3 Results	26
4-4 Conclusions	31
5 Complex Data Results	33
6 Discussion and Conclusions	35
Bibliography	37
A The back of the thesis	39
A-1 An appendix section	39
A-1-1 An appendix subsection with C++ Lisitng	39
A-1-2 A MATLAB Listing	39
B Yet another appendix	41
B-1 Another test section	41

List of Figures

2-1	Illustration of the data matrix, \mathbf{P} , by van Groenestijn (2010). <i>Left:</i> The signal generated at the source position, x_s , is measured at receiver position, x_r , as a function of time, t . Thus, the discretized data is saved in a cube, $\mathbf{p}(t, x_r, x_s)$. <i>Right:</i> The cube on the right equals the left cube after a Fourier transform with respect to time. Each frequency slice of the right cube represents the data matrix, \mathbf{P} .	4
2-2	Illustration of the data matrix, \mathbf{P} , by Mahdad et al. (2011). The dotted lines indicate directions of common gathers.	4
2-3	A conventional source matrix, \mathbf{S} , is transformed to a blended source matrix, \mathbf{S}_{bl} , by applying the blending matrix, Γ . Each star represents one shot, and the gray scale of the stars represents the relative firing time.	5
2-4	Blended shot gather of two shots. The right shot is fired 120 ms after the left shot.	6
2-5	Flowchart belonging to the deblending method of Mahdad et al. (2011).	7
2-6	Pseudo-deblended data, \mathbf{P}_{ps} , sorted in common shot gathers (a,b) and in a common receiver gather (c). The pseudo-deblended data of the right shot (a) and the left shot (b,c) were shifted by different time delays. The overlapping sources map in the pseudo-deblended shot gathers as coherent events, while they map as incoherent spikes in the pseudo-deblended receiver gather.	8
2-7	(a) Pseudo-deblended receiver gather. The subfigures (b)-(f) illustrate each step of the deblending algorithm. For better visibility examples from the 5 th iteration are chosen. (b) f - k -spectrum before (top) and after (bottom) f - k -filtering, (c) f - k -filtered common receiver gather, (d) after thresholding, (e) estimated blending noise, (f) estimated data.	9
2-8		10
2-9	Common receiver gather of the estimated data after 1, 5, 10, 15, 20 and 25 iterations.	12
3-1	Illustration of the matrix product, $\Gamma\Gamma^H$. In this notation Δt_k refers to the phase shift of the shot k , and Δt_{kl} refers to the phase shift between the shots k and l , $\Delta t_{kl} = \Delta t_k - \Delta t_l$.	14
3-2	Comparison of the pseudo-deblended receiver gather for (a) constant firing time delays of 100 ms, and (b) random firing time delays between 0 ms and 100 ms. (c) and (d) show the f - k -spectra of (a) and (b) respectively.	15

3-3	The blending matrix, Γ , is obtained by interchanging the 3 rd and 4 th row of the blending matrix in Figure 3-1. In acquisition this is equivalent to moving shot 3 to experiment 2, and shot 4 to experiment 1. A random permutation of the rows of the blending matrix spreads the off-diagonal elements of the matrix product, $\Gamma\Gamma^H$. The elements are not assembled on the sub-diagonals anymore.	15
3-4	(a) shows an synthetic unblended common receiver gather. The data are blended with a (b) spatially incoherent, (c) temporally incoherent, and (d) mixed incoherent blending pattern. The respective pseudo-deblended data (left) and deblended data (right) are shown in (b) to (d).	17
3-5	Illustration of the sub-diagonal elements in the complex number plane. The elements have unit length and variable phase. The absolute value of their sum depends on the phase coherency of the elements.	18
3-6	Illustration of the absolute sub-diagonal sums, $M(d, \omega)$, for the frequency slice $f = 25$ Hz. (a) refers to the blending matrix Γ_{coh} with constant firing time delays. (b) refers to the blending matrix Γ_{ran} with random firing time delays.	19
3-7	Deblending quality as a function of incoherency for a constant maximum firing time delay.	20
3-8	The 3 suggested blending patterns are simulated with maximum firing time delays between 40 ms and 400 ms. The quality factors are computed with respect to the unblended data and illustrated as a function of the maximum firing time delay.	21
4-1	Illustration of the data matrix \mathbf{P} for 3D data (van Dedem, 2002). y_r and y_s represent the inline receiver and shot positions. x_r and x_s represent the crossline receiver and shot positions. Each row refers to a 3D common receiver gather and each column to a 3D common shot gather. A sub-matrix with fixed receiver and source inline positions (y_r, y_s) is equivalent to a data matrix for 2D acquisition.	24
4-2	(a) 3D common receiver gather with crossline (x) and inline (y) sources (3D view). (b) Resorted data set. Individual crossline sections are plotted next to each other in 2D. For visibility both subfigures only show a reduced part of the data. This view is called 3D CRG 2D view.	25
4-3	Illustration of the blending matrix, Γ , for 3D acquisition. (a) At each of the Ns_y inline position the crossline sources (x direction) are blended. Each of these 2D blending processes is described by a 2D blending matrix, which has as many rows as there are crossline sources, Ns_x . (b) The 2D blending matrices are assembled in a single 3D blending matrix, Γ , which has Ns_x by Ns_y rows.	25
4-4	Illustration of the 3D f - k_x - k_y -filter. (a) is a 40 Hz frequency slice of the f - k_x - k_y spectrum of the data in Figure 4-2. k_x and k_y refer to the crossline and inline wavenumber respectively. (b) is a 40 Hz frequency slice of the f - k_x - k_y mask, where the white area equals 1 and the black area is 0. (c) shows the 40 Hz frequency slice of (a) sorted in a 3D cube. The red cone represents the edge of the 3D f - k_x - k_y filter mask. (d) and (e) display the f - k_x - k_y data spectrum and mask sorted according to section 4-1, i.e. each sub-cone refers to one inline wavenumber. Note that due to the sorting the wavenumber axis is a mix of crossline and inline wavenumbers. For this reason the wavenumber axis has no labels.	27
4-5	2D f - k_x -filter for 3D data. (a) shows a 40 Hz frequency slice of the f - k_x - k_y spectrum, where the white area equals 1 and the black area is 0. Note that the filter is not affecting the inline wavenumbers k_y . (b) illustrates the 2D f - k_x -filter sorted according to section 4-1. Each cone represents a 2D f - k -filter for a single inline wavenumber.	28

List of Tables

Acronyms

DUT Delft University of Technology

Chapter 1

Introduction

Chapter 2

Theory

This chapter describes the theory behind blending and deblending. First the detail hiding operator notation is explained. This notation is used to describe the forward model of seismic data. By introducing the blending operator the forward model is extended to the blended case. Next, the deblending method presented in [Mahdad et al. \(2011\)](#) is discussed to illustrate some of the concepts used in this thesis.

2-1 The Forward Model of Blending

2-1-1 Conventional Seismic Data

In the detail hiding operator notation ([Berkhout, 1982](#)) the recorded signal is considered discrete in terms of time t , receiver position x_r , and source position x_s . Thus, the measurements can be organized in a cube, $\mathbf{p}(t, x_r, x_s)$, (see Figure 2-1). Each frequency slice of this new cube represents the data matrix, \mathbf{P} .

In the data matrix, \mathbf{P} , each column corresponds to a monochromatic common shot gather (see Figure 2-2), each row to a monochromatic common receiver gather, each diagonal to a monochromatic common offset gather, and each anti-diagonal to a monochromatic common midpoint gather.

According to the seismic forward model of [Berkhout \(1982\)](#) the data matrix, \mathbf{P} , can be represented by a matrix multiplication of the source matrix, \mathbf{S} , the impulse response of the earth, \mathbf{X} , and the receiver matrix, \mathbf{D} :

$$\mathbf{P} = \mathbf{D} \mathbf{X} \mathbf{S}. \quad (2-1)$$

In the source matrix, \mathbf{S} , both rows and columns represent shot positions (see Figure 2-3). Thus, \mathbf{S} is a diagonal matrix. Each diagonal element s_{ii} captures one frequency component of the source signature injected in the earth at the position $x_s = x_i$. By applying a Fourier

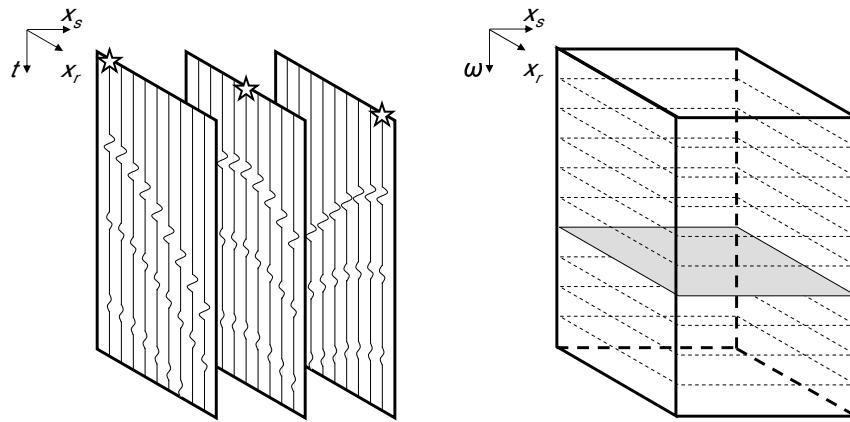


Figure 2-1: Illustration of the data matrix, \mathbf{P} , by [van Groenestijn \(2010\)](#). *Left:* The signal generated at the source position, x_s , is measured at receiver position, x_r , as a function of time, t . Thus, the discretized data is saved in a cube, $\mathbf{p}(t, x_r, x_s)$. *Right:* The cube on the right equals the left cube after a Fourier transform with respect to time. Each frequency slice of the right cube represents the data matrix, \mathbf{P} .

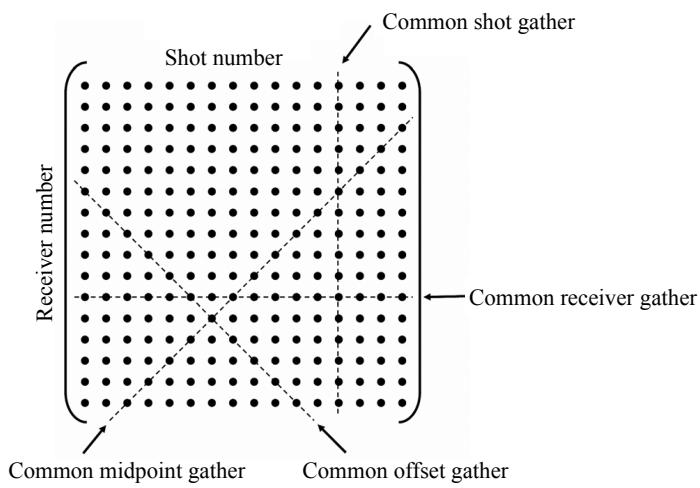


Figure 2-2: Illustration of the data matrix, \mathbf{P} , by [Mahdad et al. \(2011\)](#). The dotted lines indicate directions of common gathers.

$$\begin{array}{|c|c|c|c|} \hline
 \star & 0 & 0 & 0 \\ \hline
 0 & \star & 0 & 0 \\ \hline
 0 & 0 & \star & 0 \\ \hline
 0 & 0 & 0 & \star \\ \hline
 \end{array}
 \cdot
 \begin{array}{|c|c|} \hline
 1 & 0 \\ \hline
 0 & 1 \\ \hline
 e^{-j\omega\Delta t_1} & 0 \\ \hline
 0 & e^{-j\omega\Delta t_2} \\ \hline
 \end{array}
 =
 \begin{array}{|c|c|} \hline
 \star & 0 \\ \hline
 0 & \star \\ \hline
 \star & 0 \\ \hline
 0 & \star \\ \hline
 \end{array}$$

S **Γ** **S_{bl}**

Figure 2-3: A conventional source matrix, \mathbf{S} , is transformed to a blended source matrix, \mathbf{S}_{bl} , by applying the blending matrix, $\mathbf{\Gamma}$. Each star represents one shot, and the gray scale of the stars represents the relative firing time.

transform to all frequency components of the element s_{ii} the source signature as a function of time is obtained.

The impulse response of the earth, \mathbf{X} , describes how an impulse at the source location, x_s , is transformed in the earth into the signal at the receiver location, x_r .

The receiver matrix, \mathbf{D} , converts the seismic wavefield at the receiver location, x_r , to the recorded signal. This includes adding the receiver ghost.

In practice, one tries to retrieve the unknown earth response, \mathbf{X} , from the data, \mathbf{P} , by removing \mathbf{S} (signature) and \mathbf{D} (receiver deghosting).

2-1-2 Blended Seismic Data

In blended acquisition the recorded events belonging to different shots overlap, as shown in the shot gather in Figure 2-4.

Blending can be captured in the forward model by introducing a blending matrix, $\mathbf{\Gamma}$, which transforms the source matrix, \mathbf{S} , into a blended source matrix, \mathbf{S}_{bl} ,

$$\mathbf{S}_{bl} = \mathbf{S} \mathbf{\Gamma}. \quad (2-2)$$

Figure 2-3 shows the structure of $\mathbf{\Gamma}$; each row of $\mathbf{\Gamma}$ represents one source, and each column of $\mathbf{\Gamma}$ represents one experiment.

The blending matrix captures the physics of a blended acquisition as follows: An element γ_{ij} of the blending matrix includes a source i and an experiment j . If the source i is not fired in the j^{th} experiment γ_{ij} is zero. If it is fired, source i has a relative amplitude A_{ij} and a relative time delay Δt_{ij} with respect to the first source fired in the j^{th} experiment;

$$\gamma_{ij} = A_{ij} e^{-j\omega\Delta t_{ij}}. \quad (2-3)$$

Thus, the blending matrix selects specific sources from the source matrix and superimposes them as visualized in Figure 2-3. From Figure 2-3 it also becomes clear that both the blending

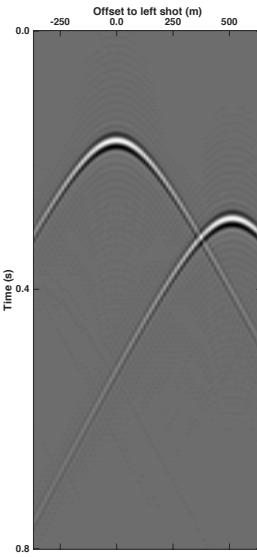


Figure 2-4: Blended shot gather of two shots. The right shot is fired 120 ms after the left shot.

matrix, Γ , and the blended source matrix, \mathbf{S}_{bl} , have more rows than columns, i.e. there are more sources than experiments. Thus, the acquisition is done in less time.

In the case of source blending the receiver matrix, \mathbf{D} , is not influenced. Of course, the earth impulse response, \mathbf{X} , is independent of the acquisition design. Hence, the blended data can be written as;

$$\mathbf{P}_{bl} = \mathbf{D} \mathbf{X} \mathbf{S}_{bl} = \mathbf{D} \mathbf{X} \mathbf{S} \Gamma = \mathbf{P} \Gamma. \quad (2-4)$$

Note that, the blended data matrix, \mathbf{P}_{bl} , also has less columns, i.e. less experiments, than the unblended data matrix, \mathbf{P} .

2-2 Deblending

Before removing the receiver matrix, \mathbf{D} , and the source matrix, \mathbf{S} , one must remove the blending matrix, Γ , from the blended data, \mathbf{P}_{bl} . This process is called deblending.

The deblending method presented in this thesis builds on the method of [Mahdad et al. \(2011\)](#). Therefore, the this method is described in great detail.

The basic workflow of the Mahdad method is summarized in Figure 2-5 and will be explained step by step in the following subsections.

2-2-1 Pseudo-Deblending

Unfortunately, the inverse problem of equation 2-4 is underdetermined, which means that there is not a unique solution for the unblended data, \mathbf{P} . Thus, additional constraints are

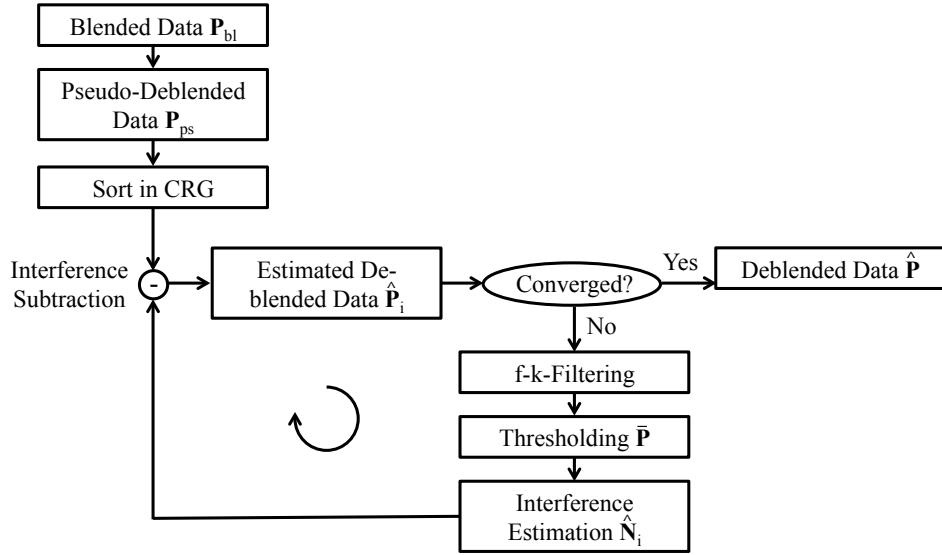


Figure 2-5: Flowchart belonging to the deblending method of Mahdad et al. (2011).

required to deblend the data, which are; (1) sparsity of the signal in the x - t -domain and (2) coherency of the signal in the f - k -domain.

The first estimate of the unblended data matrix, \mathbf{P} , is obtained by pseudo-deblending;

$$\mathbf{P}_{ps} = \mathbf{P}_{bl} \boldsymbol{\Gamma}^H. \quad (2-5)$$

Pseudo-deblending copies the blended data to the locations of all shots present in the blended shot and shifts them upward in time to compensate for the time delay. For example, Figure 2-6a and 2-6b shows the two pseudo-deblended shot gathers of the blended data in Figure 2-4. Note that the pseudo-deblended data, \mathbf{P}_{ps} , have the same size as \mathbf{P} .

2-2-2 Common Receiver Gather

In Figure 2-6a and 2-6b the interfering shot is coherent. By transforming the data to another domain, e.g. to the common receiver domain, the interfering shot becomes incoherent and is visible as spiky noise (see Figure 2-6c). Therefore, the interfering shot can be attenuated with a noise filter.

2-2-3 Iterative Estimation of Blending Noise

In an ideal case the noise generated by the interfering shots present in the pseudo-deblended data, the so called blending noise, is calculated with the unblended data,

$$\mathbf{N} = \mathbf{P}_{bl} \boldsymbol{\Gamma}^H - \mathbf{P} = \mathbf{P}_{ps} - \mathbf{P}. \quad (2-6)$$



Figure 2-6: Pseudo-deblended data, \mathbf{P}_{ps} , sorted in common shot gathers (a,b) and in a common receiver gather (c). The pseudo-deblended data of the right shot (a) and the left shot (b,c) were shifted by different time delays. The overlapping sources map in the pseudo-deblended shot gathers as coherent events, while they map as incoherent spikes in the pseudo-deblended receiver gather.

Obviously, in practice the unblended data are unknown and are estimated by adding extra constraints. The loop shown in Figure 2-5 applies the constraints to reduce the blending noise iteratively until the solution is obtained.

In the following all the quantities which are estimated are indicated with a hat. The steps of the iterative blending noise estimation are illustrated in Figure 2-7.

f-k-Filtering

One of the constraints is coherency, i.e. by assuming the blending noise in Figure 2-7a is incoherent it can be removed. For this purpose the data are transformed from the space time to the wavenumber frequency domain where the spiky noise spreads over all wavenumber and frequency components (see Figure 2-7b, top).

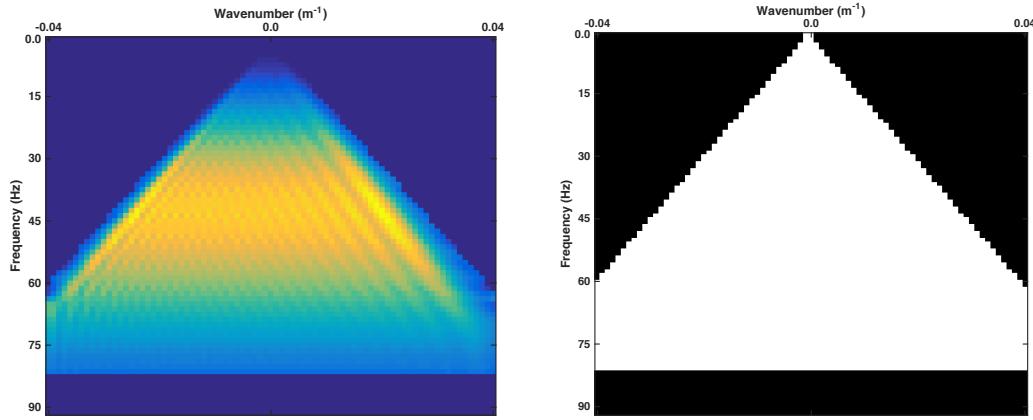
The unblended data would map in the *f-k*-domain in a cone (see Figure 2-8a). The minimum wavefield velocity present in the subsurface, v_{min} , determines the slope of the cone. This means that for a given frequency, f , the maximum wavenumber inside the cone, k_{max} , is defined as;

$$k_{max} = \frac{f}{v_{min}}. \quad (2-7)$$

In the marine case the minimum velocity is usually the water velocity, $v_w = 1500 \text{ m s}^{-1}$.



Figure 2-7: (a) Pseudo-deblended receiver gather. The subfigures (b)-(f) illustrate each step of the deblending algorithm. For better visibility examples from the 5th iteration are chosen. (b) $f\text{-}k$ -spectrum before (top) and after (bottom) $f\text{-}k$ -filtering, (c) $f\text{-}k$ -filtered common receiver gather, (d) after thresholding, (e) estimated blending noise, (f) estimated data.



(a) $f\text{-}k$ -spectrum of an unblended common shot gather.

(b) The $f\text{-}k$ -mask is determined by the minimum signal velocity in the subsurface. The white area of the filter equals one, the black area equals zero. Thus, the filter removes data which are mapped outside of the white signal cone.

Figure 2-8

A 2D $f\text{-}k$ -filter can be designed that removes all elements outside the cone (see Figure 2-8b). The $f\text{-}k$ -filter removes the part of the blending noise, which maps outside of the signal cone (see Figure 2-7b). Thus, after transforming the data back to the space time domain the amplitudes of the spiky noise are attenuated (see Figure 2-7c).

Note that $f\text{-}k$ -filtering can only reduce spatially unaliased blending noise. In Figure 2-8b the highest spatially unaliased frequency is defined by the point where the white cone intersects with the frequency axis, i.e. at 60 Hz. The spatially aliased blending noise will pass the $f\text{-}k$ -filter and will be reduced afterwards by thresholding.

The high cut frequency of the $f\text{-}k$ -mask, $f_{cut} = 80$ Hz, is set according to the highest frequency components in the data.

Thresholding

The second constraint for the estimation of the unblended data is sparsity of the signal in the space time domain.

After $f\text{-}k$ -filtering the spiky noise is attenuated (see Figure 2-7c). Consequently, the signal amplitudes are now stronger than the noise amplitudes. This allows to define a threshold in the $x\text{-}t$ domain, which is larger than the attenuated noise amplitudes and smaller than the highest signal amplitudes. Only amplitudes above the threshold are picked (see Figure 2-7d).

Interference Estimation

The resulting thresholded data, $\bar{\mathbf{P}}$, is used to predict the blending noise;

$$\hat{\mathbf{N}}_i = \bar{\mathbf{P}} (\mathbf{\Gamma} \mathbf{\Gamma}^H - \mathbf{I}), \quad (2-8)$$

which is illustrated in Figure 2-7e.

Blending Noise Subtraction

The estimate of the unblended data matrix $\hat{\mathbf{P}}_i$ is updated by subtracting the estimated blending noise from the pseudo-deblended data,

$$\hat{\mathbf{P}}_{i+1} = \mathbf{P}_{ps} - \hat{\mathbf{N}}_i, \quad (2-9)$$

which is shown in Figure 2-7f.

This process is repeated iteratively till convergence is reached. In this context convergence can be defined as the point where the difference $| \hat{\mathbf{P}}_{i+1} - \hat{\mathbf{P}}_i |$ drops below a predefined limit. Alternatively, one can set a maximum number of iterations.

Figure 2-9 shows the estimate of the unblended data for increasing iterations. At each iteration the blending noise is attenuated further, such that the threshold can be lowered. Hence, the predicted blending noise increases and approaches the true blending noise. The two blended shots are successively deblended.

Note that, f - k -filtering lowers the noise level by removing spatially unaliased blending noise. Next, the lowered noise level enables thresholding to reduce spatially aliased blending noise. Thus, the combination of f - k -filtering and thresholding is very powerful.



Figure 2-9: Common receiver gather of the estimated data after 1, 5, 10, 15, 20 and 25 iterations.

Chapter 3

Incoherency

In this chapter the blending operator is analyzed in greater detail. Then a measure for incoherency will be introduced. Finally, it will be discussed how deblending is influenced by incoherency and the so called maximum firing time delay.

3-1 Analysis of the Blending Matrix

In order to optimize the blended acquisition design, one must understand the properties of the blending matrix, Γ , and its influence on the deblending performance.

The blending matrix, Γ , determines the pseudo-deblended data,

$$\mathbf{P}_{ps} = \mathbf{P}\Gamma\Gamma^H, \quad (3-1)$$

which are a superposition of the unblended data, \mathbf{P} , and the blending noise, \mathbf{N} ,

$$\mathbf{P}_{ps} = \mathbf{P} + \mathbf{N} = \mathbf{P}\mathbf{I} + \mathbf{P}(\mathbf{\Gamma}\mathbf{\Gamma}^H - \mathbf{I}). \quad (3-2)$$

The more incoherent the blending noise, \mathbf{N} , the better it can be removed by noise filters.

In the following the effect of the blending matrix, Γ , on the pseudo-deblended data, \mathbf{P}_{ps} , is analyzed. For simplicity, it is assumed that all shots are equal in strength and fire the same signature into the earth. It is also assumed that each shot is fired only once, unlike e.g. the shot repetition case (Wu, 2014). This means that the blending matrix, Γ , only contains phase shift terms, $e^{-j\omega\Delta t}$, or zeros.

Each row of Γ represents a shot k , and each column of Γ^H represents a shot l with a complex conjugated phase term (see Figure 3-1). Hence, each element g_{kl} of the matrix $\mathbf{\Gamma}\mathbf{\Gamma}^H$ is the dot product between the k^{th} shot and the complex conjugate of the l^{th} shot.

Consequently, an element g_{kl} of the matrix product, $\mathbf{\Gamma}\mathbf{\Gamma}^H$, represents the overlap of the shots k and l for all experiments. The main diagonal of $\mathbf{\Gamma}\mathbf{\Gamma}^H$ refers to the overlap of each shot with

$$\begin{array}{c}
 \text{shot k} \\
 \begin{array}{|c|c|} \hline e^{-j\omega\Delta t_i} & 0 \\ \hline 0 & e^{-j\omega\Delta t_k} \\ \hline e^{-j\omega\Delta t_j} & 0 \\ \hline 0 & e^{-j\omega\Delta t_l} \\ \hline \end{array}
 \end{array}
 \cdot
 \begin{array}{c}
 \text{shots} \\
 \begin{array}{|c|c|c|c|} \hline e^{+j\omega\Delta t_i} & 0 & e^{+j\omega\Delta t_j} & 0 \\ \hline 0 & e^{+j\omega\Delta t_k} & 0 & e^{+j\omega\Delta t_l} \\ \hline \end{array}
 \end{array}
 =
 \begin{array}{c}
 \begin{array}{|c|c|c|c|} \hline 1 & 0 & e^{-j\omega\Delta t_{ij}} & 0 \\ \hline 0 & 1 & 0 & e^{-j\omega\Delta t_{kl}} \\ \hline e^{-j\omega\Delta t_{ji}} & 0 & 1 & 0 \\ \hline 0 & e^{-j\omega\Delta t_{lk}} & 0 & 1 \\ \hline \end{array}
 \end{array}$$

Γ Γ^H $\Gamma\Gamma^H$

Figure 3-1: Illustration of the matrix product, $\Gamma\Gamma^H$. In this notation Δt_k refers to the phase shift of the shot k , and Δt_{kl} refers to the phase shift between the shots k and l , $\Delta t_{kl} = \Delta t_k - \Delta t_l$.

itself, which of course is perfect and therefore equal to 1. The off diagonal elements of $\Gamma\Gamma^H$ are either 0 if the associated shots do not overlap, or contain a phase shift, $e^{-j\omega\Delta t_{kl}}$.

Temporal incoherency

In the following the term "sub-diagonal" will be used to refer to an arbitrary diagonal of the matrix $\Gamma\Gamma^H$. For example, the d^{th} sub-diagonal includes all the matrix elements g_{ij} , which fulfill the condition; $j - i = d$.

In equation 3-2 the main diagonal elements of $\Gamma\Gamma^H$ copy the data matrix, \mathbf{P} , while the off-diagonal elements create the blending noise, \mathbf{N} . In case of constant firing time delays the elements along a sub-diagonal d all have the same phase. This means that the sub-diagonal elements will shift the columns of the data matrix and apply a constant phase shift to each of them resulting in the pseudo-deblended receiver gather shown in Figure 3-2a. Instead if firing times are not constant but random the elements g_{ij} along a sub-diagonal d will have different phases. Consequently, they will shift the columns of the data matrix and distort the phase of each column (see Figure 3-2b).

Figure 3-2c and 3-2d display the f - k -spectra of the pseudo-blended data for constant firing time delays and random firing time delays respectively. In the case of constant firing time delays all of the energy maps in the signal cone. In the case of random firing time delays a significant part of the energy maps outside of the signal cone. From Figure 3-2c and 3-2d it is clear that the coherency constraint presented in section 2-2-3 cannot work with constant firing time delays, but needs the random firing time delays.

In this thesis the random firing time delays along a sub-diagonal are referred to as temporal incoherency.

Spatial incoherency

Of course, the degree of incoherency of the blending noise, \mathbf{N} , also depends on whether the shot positions of shots blended in an experiment are selected randomly, or in a spatially coherent pattern. For example, one expects the blending noise to be more incoherent if in

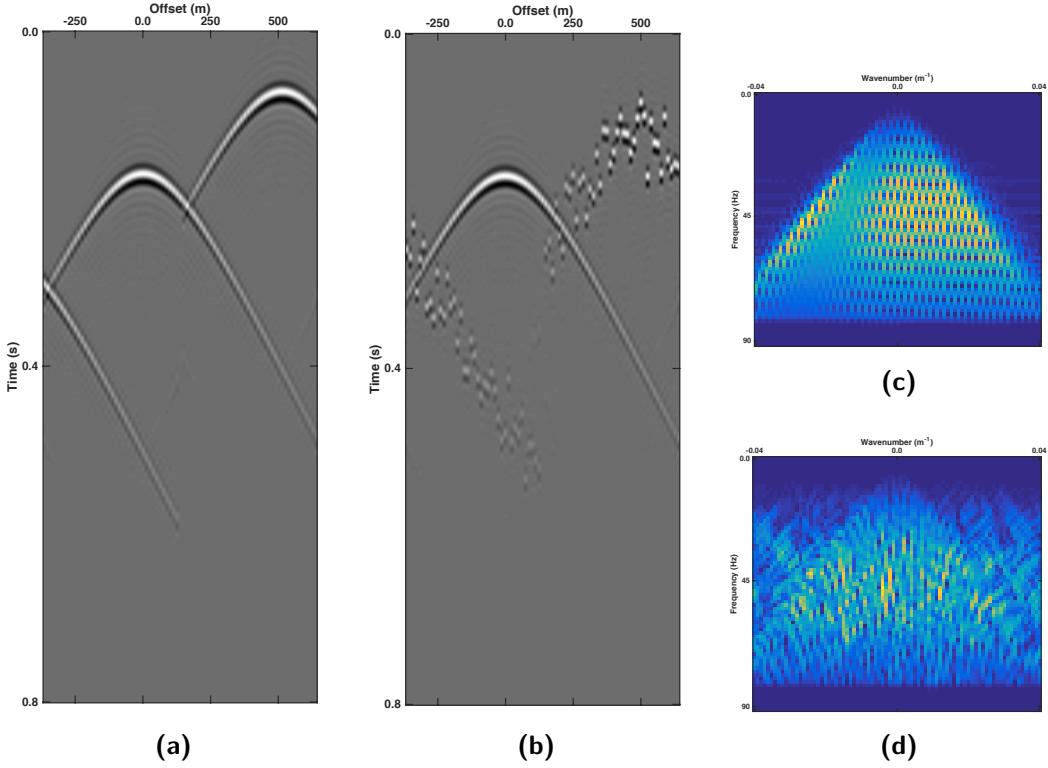


Figure 3-2: Comparison of the pseudo-deblended receiver gather for (a) constant firing time delays of 100 ms, and (b) random firing time delays between 0 ms and 100 ms. (c) and (d) show the f - k -spectra of (a) and (b) respectively.

$$\begin{array}{c}
 \begin{array}{|c|c|} \hline
 e^{-j\omega\Delta t_i} & 0 \\ \hline
 0 & e^{-j\omega\Delta t_k} \\ \hline
 0 & e^{-j\omega\Delta t_l} \\ \hline
 e^{-j\omega\Delta t_j} & 0 \\ \hline
 \end{array} \cdot \begin{array}{|c|c|c|c|} \hline
 & \text{shot } k & & \\ \hline
 & * & & \\ \hline
 e^{+j\omega\Delta t_i} & 0 & 0 & e^{+j\omega\Delta t_j} \\ \hline
 0 & e^{+j\omega\Delta t_k} & e^{+j\omega\Delta t_l} & 0 \\ \hline
 \end{array} = \begin{array}{|c|c|c|c|} \hline
 1 & 0 & 0 & e^{-j\omega\Delta t_{ij}} \\ \hline
 0 & 1 & e^{-j\omega\Delta t_{kl}} & 0 \\ \hline
 0 & e^{-j\omega\Delta t_{lk}} & 1 & 0 \\ \hline
 e^{-j\omega\Delta t_{ji}} & 0 & 0 & 1 \\ \hline
 \end{array} \\
 \Gamma \qquad \qquad \qquad \Gamma^H \qquad \qquad \qquad \Gamma \Gamma^H
 \end{array}$$

Figure 3-3: The blending matrix, Γ , is obtained by interchanging the 3rd and 4th row of the blending matrix in Figure 3-1. In acquisition this is equivalent to moving shot 3 to experiment 2, and shot 4 to experiment 1. A random permutation of the rows of the blending matrix spreads the off-diagonal elements of the matrix product, $\Gamma \Gamma^H$. The elements are not assembled on the sub-diagonals anymore.

each experiment randomly selected shot positions are blended as in Figure 3-3, than if in each experiment adjacent shot positions are blended as in Figure 3-1, because the interfering shots are spread over the sub-diagonals in Figure 3-3.

In this thesis selecting random shots for an experiment is referred to as spatial incoherency.

Although examples of spatial incoherency are shown in this chapter, it has to be noted that to blend shots in a spatially incoherent fashion is not very practical in 2D acquisition. However, in chapter 4 it will be shown that for 3D blending spatial incoherency becomes feasible.

3-2 Results Spatial and Temporal Incoherency

Based on the blending matrices in Figure 3-1 and 3-3 there are three possibilities to blend the shots incoherently. First, the phase terms along the sub-diagonals can be randomly varied, i.e. the shots are blended with random firing time delays (temporal incoherency). Second, the rows of the blending matrix can be randomly permuted, i.e. one randomly selects shots for each experiment (spatial incoherency). Third, temporal and spatial incoherency can be combined (mixed incoherency), i.e. randomly selected shots are blended with random firing time delays.

These 3 blending patterns are applied to a synthetic data set. The data are a common receiver gather with 21 shots (see Figure 3-4a), which are blended in 3 experiments with 7 shots per experiment. Next, the data are deblended with the deblending algorithm of section 2-2.

The deblended receiver gathers are shown in Figure 3-4. The results suggest that only spatial incoherency is not sufficient to deblend the data (see Figure 3-4b). By introducing random firing time delays the deblended data improve significantly as shown in Figure 3-4c. A combination of both spatial and temporal incoherency enhances the deblended data further (see Figure 3-4d).

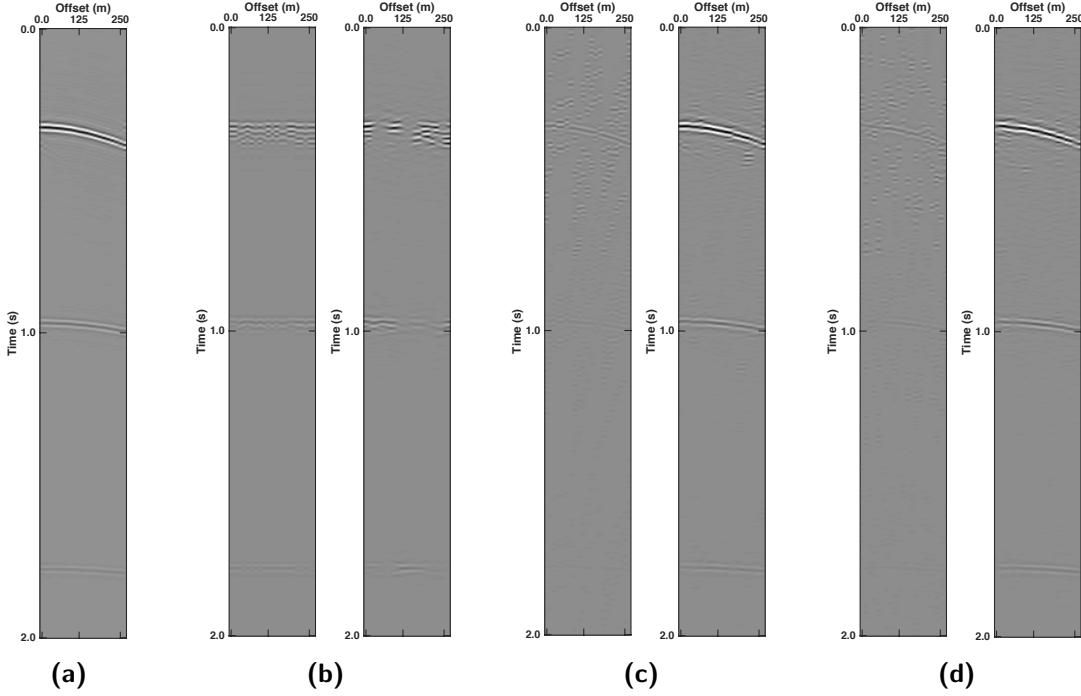


Figure 3-4: (a) shows an synthetic unblended common receiver gather. The data are blended with a (b) spatially incoherent, (c) temporally incoherent, and (d) mixed incoherent blending pattern. The respective pseudo-deblended data (left) and deblended data (right) are shown in (b) to (d).

3-3 Effect of Incoherency

This section aims to analyze how strongly the deblending result depends on the incoherency of the blended acquisition. For this purpose a measure of incoherency and deblending quality will be introduced.

Incoherency Measure

In section 3-1 it was shown that for an incoherent blending pattern the elements, $e^{-j\omega\Delta t_{kl}}$, along a sub-diagonal of the matrix product, $\mathbf{\Gamma}\mathbf{\Gamma}^H$, should be out of phase. Therefore, the phase variability of the sub-diagonal elements will be used to quantify incoherency.

The sub-diagonal elements, $e^{-j\omega\Delta t_{kl}}$, map in the complex plane on a circle with radius 1 (see Figure 3-5). Thus, the sum of the elements along the d^{th} sub-diagonal can be constructive or destructive, depending on the phase variability. The absolute value of this sum, $M(d, \omega)$, measures the incoherency of the d^{th} sub-diagonal for the frequency component, ω ;

$$M(d, \omega) = \left| \sum_{j-i=d} \mathbf{\Gamma}\mathbf{\Gamma}_{ij}^H(\omega) \right|. \quad (3-3)$$

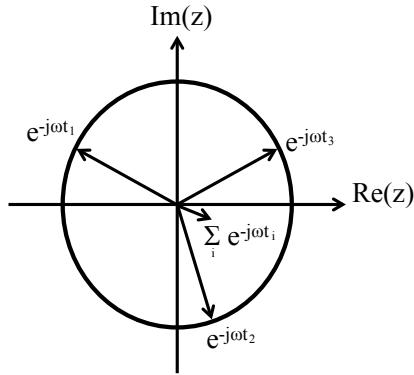


Figure 3-5: Illustration of the sub-diagonal elements in the complex number plane. The elements have unit length and variable phase. The absolute value of their sum depends on the phase coherency of the elements.

If all sub-diagonal elements are in phase the absolute value of their sum, $M(d, \omega)$, is maximized. Instead, in case of an incoherent blending pattern $M(d, \omega)$ is small for all sub-diagonals d , except for the main diagonal ($d = 0$).

The incoherency measure, μ , is introduced as;

$$\mu = \frac{(\sum_{\omega} M(d=0, \omega))^2}{\sum_{d=1-N_s}^{N_s-1} ((\sum_{\omega} M(d, \omega))^2)}, \quad (3-4)$$

where N_s is the number of sources, i.e. the matrix $\mathbf{\Gamma}\mathbf{\Gamma}^H$ has N_s rows and columns. Note that in this ratio, the numerator relates to the main diagonal only, and the denominator to all sub-diagonals.

Equation 3-4 implies that the incoherency value can vary between 0 (perfectly coherent) and 1 (perfectly incoherent).

For example, consider the two blending matrices, $\mathbf{\Gamma}_{coh}$ and $\mathbf{\Gamma}_{ran}$, which produce the pseudo-deblended receiver gathers in Figure 3-2a and 3-2b respectively. The blending matrix, $\mathbf{\Gamma}_{coh}$, uses constant firing time delays, while the blending matrix, $\mathbf{\Gamma}_{ran}$, uses random firing time delays. Figure 3-6 shows $M(d, \omega)$ for both blending matrices for the frequency slice $f = 25$ Hz. One can observe that for the incoherent blending matrix, $\mathbf{\Gamma}_{ran}$, $M(d, \omega)$ resembles a simple spike more than for the coherent blending matrix, $\mathbf{\Gamma}_{coh}$. The coherent blending matrix, $\mathbf{\Gamma}_{coh}$, yields an incoherency value of $\mu_{coh} = 67\%$, whereas the incoherent blending matrix, $\mathbf{\Gamma}_{ran}$, yields an incoherency value of $\mu_{ran} = 98\%$.

Deblending Performance Measure

The following data examples are synthetic data, i.e. the unblended data are known. Therefore, the deblending performance can be measured with the quality factor, Q , which is defined by Ibrahim and Sacchi (2015) as;

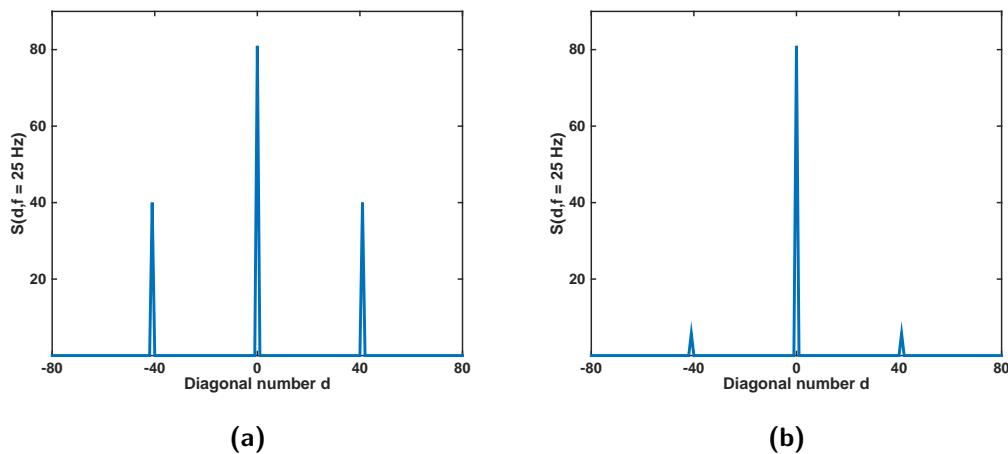


Figure 3-6: Illustration of the absolute sub-diagonal sums, $M(d, \omega)$, for the frequency slice $f = 25$ Hz. (a) refers to the blending matrix Γ_{coh} with constant firing time delays. (b) refers to the blending matrix Γ_{ran} with random firing time delays.

$$Q = 10 \cdot \log_{10} \left(\frac{\|\mathbf{P}\|_2^2}{\left\| \mathbf{P} - \hat{\mathbf{P}} \right\|_2^2} \right), \quad (3-5)$$

where \mathbf{P} are the unblended data, and $\hat{\mathbf{P}}$ the deblended data. This is similar to the definition of signal to noise ratio.

3-4 Results Incoherency

Consider again the two blending matrices, $\boldsymbol{\Gamma}_{coh}$ and $\boldsymbol{\Gamma}_{ran}$, which produced the pseudo-deblended receiver gathers in Figure 3-2a and 3-2b respectively. Both blending matrices use the same maximum firing time delay, while their incoherency values differ ($\mu_{coh} = 67\%$, $\mu_{ran} = 98\%$).

An effective blending matrix, Γ_{eff} , is created by superimposing the time delays $e^{-j\omega\Delta t_{ij}}$ of the blending matrices Γ_{coh} and Γ_{ran} ;

$$\Delta t_{eff} = a \cdot \Delta t_{ij,ran} + (1 - a) \cdot \Delta t_{ij,coh}, \quad a \in [0, 1]. \quad (3-6)$$

The maximum firing time delay of the effective blending matrix, Γ_{eff} , is constant while the incoherency varies with changing a . The resulting quality factors are shown as a function of the incoherency in Figure 3-7.

One can observe that the deblending quality increases with increasing incoherency.

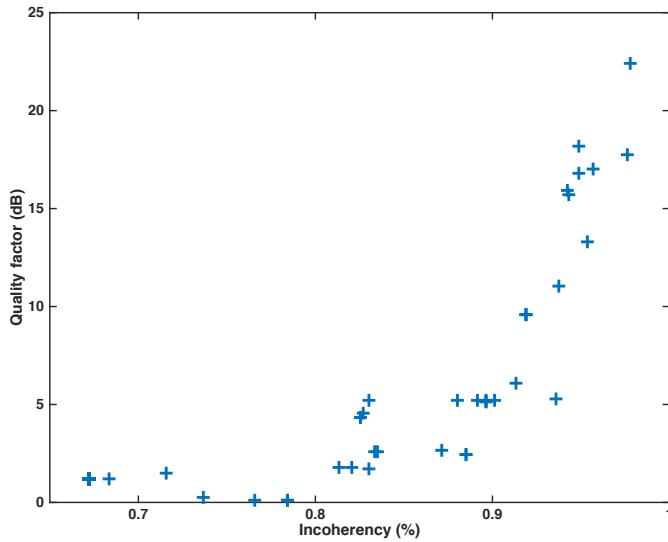


Figure 3-7: Deblending quality as a function of incoherency for a constant maximum firing time delay.

3-5 Effect of Maximum Firing Time Delay

Another control factor of the deblending quality is the maximum firing time delay. In an extreme case of infinitely long maximum firing time delay the acquisition is not blended any more, and the deblending result is perfect. Hence, increasing maximum firing time delays are expected to enhance the deblending quality. Of course, they require more acquisition time.

3-6 Results Maximum Firing Time Delay

The three suggested blending patterns namely temporal, spatial and mixed incoherency are applied to synthetic data with varying maximum firing time delays.

Figure 3-8 shows the quality factors for the three blending patterns as a function of maximum firing time delay. Note that for a fixed maximum firing time delay and a specific blending pattern the firing time delays are generated with a random number generator. Consequently, the resulting quality factor varies depending on the variation of the random number series. For this reason several blending matrices are generated for each maximum firing time delay and each blending pattern. The resulting quality factors are averaged.

According to Figure 3-8 the spatially incoherent blending pattern yields a constant deblending quality independent of the maximum firing time delay. This is expected because it blends the sources without time delay. The deblending quality provided by the other two blending patterns continuously enhances with increasing maximum firing time delay. The difference in deblending quality between temporal and mixed blending patterns seems to be independent of the maximum firing time delay.

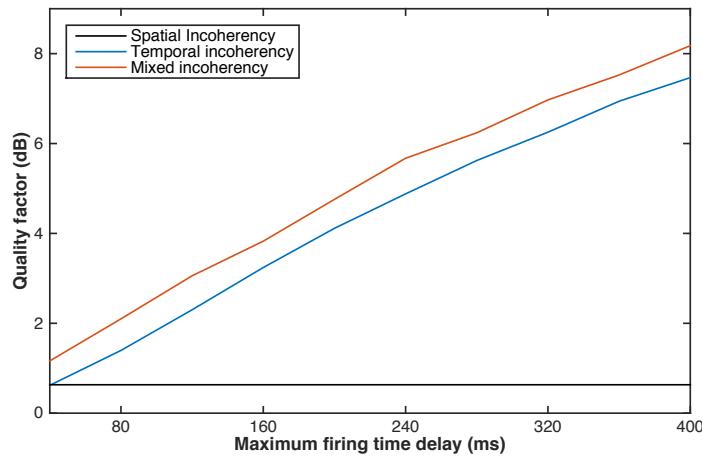


Figure 3-8: The 3 suggested blending patterns are simulated with maximum firing time delays between 40 ms and 400 ms. The quality factors are computed with respect to the unblended data and illustrated as a function of the maximum firing time delay.

3-7 Conclusions

In practice, the maximum firing time delay is given by the available acquisition time. Figure 3-7 demonstrates that for a fixed maximum firing time delay the deblending quality can be optimized by choosing a blending pattern with maximum incoherency. The maximum achievable deblending quality for a fixed maximum firing time delay can be estimated from Figure 3-8.

Chapter 4

Crossline Deblending (3D)

This thesis suggests to blend crossline sources. By combining several cross-lines one effectively blends sources in 3D.

The deblending method of [Mahdad et al. \(2011\)](#) described in section [2-2](#) is designed for 2D blended data. In this chapter I will explain how each step of the Mahdad method can be applied to 3D data as well, and I will demonstrate its performance.

First, the data sorting will be modified such that the blended 3D data can be described using the same forward model as in section [2-1](#). The presented data sorting will allow to maintain all other steps of the deblending algorithm of [Mahdad et al. \(2011\)](#) unchanged. Second, the $f\text{-}k$ -filter will be extended to an $f\text{-}k_x\text{-}k_y$ -filter to remove noise in both crossline and inline direction.

4-1 Data Sorting

Data Matrix

In 3D acquisition the sources and receivers are distributed on a 2D surface. Thus, their locations are defined by their crossline and inline positions, (x, y) . Each data point which is measured by a source receiver pair at a specific time is therefore described by 5 coordinates, time t , receiver crossline and inline position (x_r, y_r) , and source crossline and inline position (x_s, y_s) .

Similar as in section [2-1](#) the 5D data "cube" will be again reorganized in a 2D data matrix according to [van Dedem \(2002\)](#) (see Figure [4-1](#)). For this data sorting a 1D Fourier transform with respect to time is performed and a 4D frequency "slice" is selected.

The 4D "slice" is sorted in a 2D data matrix, \mathbf{P} , with as many rows as receivers and as many columns as shots. The total number of shots is obtained by multiplying the number of shots fired in each crossline and the number of shots fired in each inline. The total number of receivers is obtained likewise. Assume there are Ns_x shots per crossline. The shots of

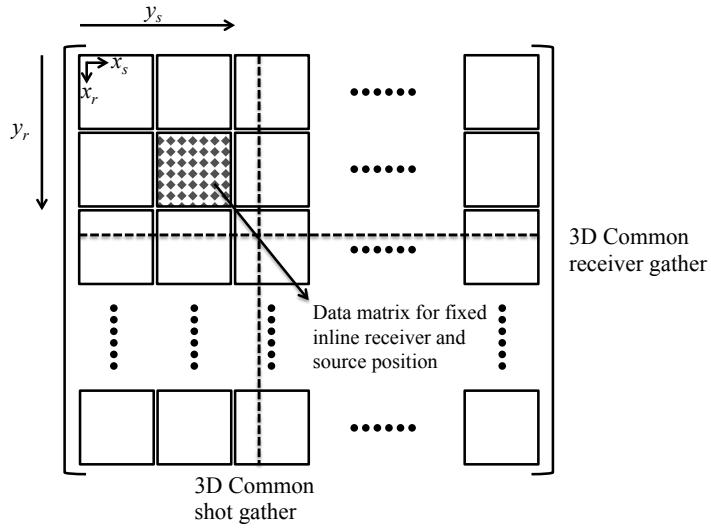


Figure 4-1: Illustration of the data matrix \mathbf{P} for 3D data (van Dedem, 2002). y_r and y_s represent the inline receiver and shot positions. x_r and x_s represent the crossline receiver and shot positions. Each row refers to a 3D common receiver gather and each column to a 3D common shot gather. A sub-matrix with fixed receiver and source inline positions (y_r, y_s) is equivalent to a data matrix for 2D acquisition.

the first crossline are assigned to the first Ns_x columns of the data matrix, the shots of the second crossline are assigned to the next Ns_x columns of the data matrix, etc. The receivers are sorted in the rows of the data matrix in analogy.

One row in the data matrix, \mathbf{P} , in Figure 4-1 represents a 3D common receiver gather. The data of this 3D common receiver gather are shown in Figure 4-2a in a 3D view, where the coordinates, x and y , indicate the crossline and inline shot position respectively. For the described data sorting individual crossline slices are extracted from this data cube and assembled next to each other in a data matrix as shown in Figure 4-2b. This view will be referred to as 3D CRG 2D view. Each hyperbolic event refers to the response of the shots of one crossline.

Blending matrix

The blending matrix for 3D is build in a similar fashion as the data matrix in 3D. As described in section 3-1 each row of the blending matrix, $\mathbf{\Gamma}$, captures one shot. For extension to 3D the shots of the first crossline are placed in the top Ns_x rows of the blending matrix, followed by the shots of the second crossline etc. (see Figure 4-3). The elements in the j^{th} column of the blending matrix, $\mathbf{\Gamma}$, select the shots which are blended in the j^{th} experiment. For example, the first column of the blending matrix in Figure 4-3b describes that in the first experiment shots (rows) 1 and 3 are blended with a time delay of Δt_1 .

With the new data and blending matrix sorting one can apply deblending to 3D data in the same way as presented in section 2-1 and section 2-2. Note that unlike for 2D blending spatially incoherent blending patterns are practical for 3D blended acquisition.

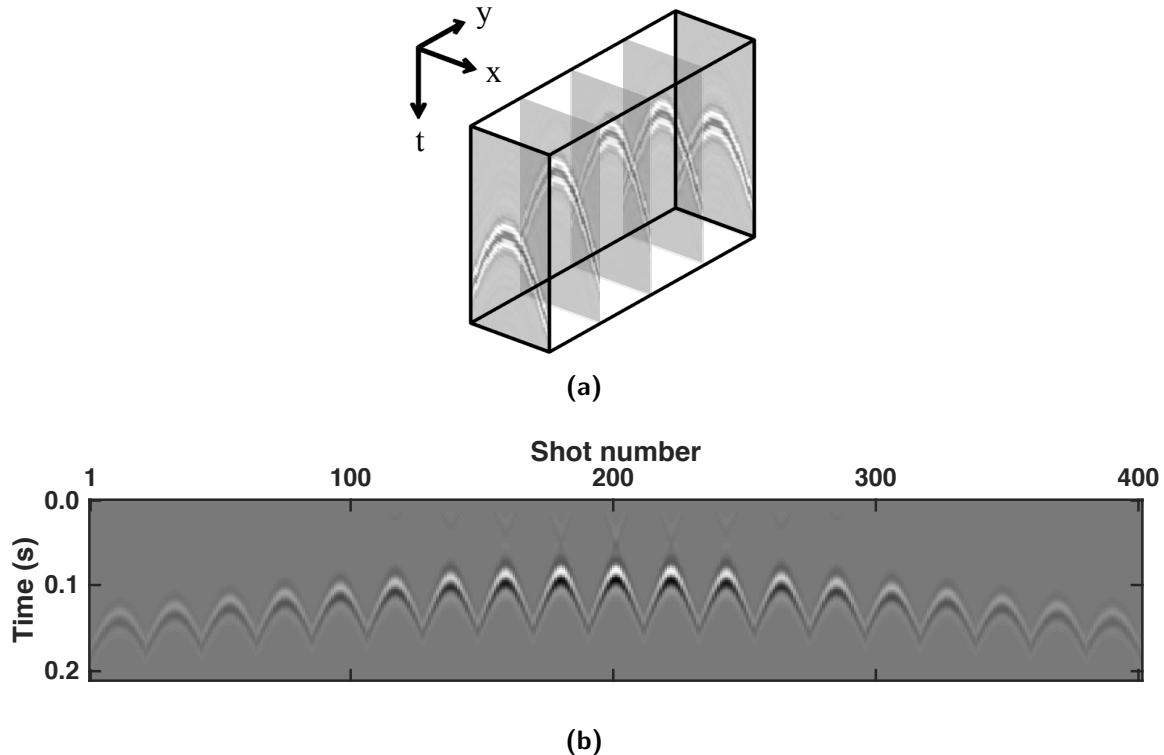


Figure 4-2: (a) 3D common receiver gather with crossline (x) and inline (y) sources (3D view).
 (b) Resorted data set. Individual crossline sections are plotted next to each other in 2D. For visibility both subfigures only show a reduced part of the data. This view is called 3D CRG 2D view.

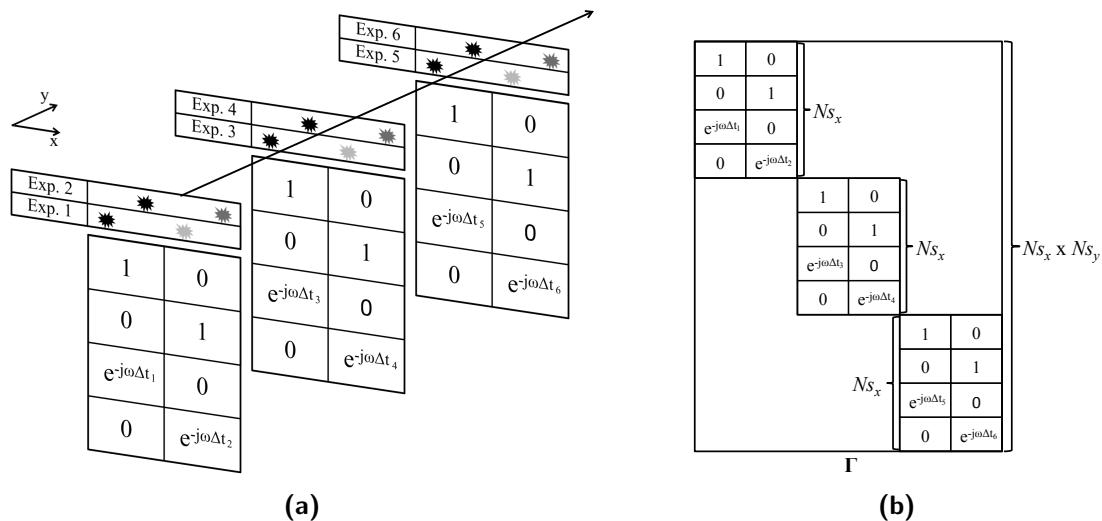


Figure 4-3: Illustration of the blending matrix, Γ , for 3D acquisition. (a) At each of the N_{s_y} inline position the crossline sources (x direction) are blended. Each of these 2D blending processes is described by a 2D blending matrix, which has as many rows as there are crossline sources, N_{s_x} . (b) The 2D blending matrices are assembled in a single 3D blending matrix, Γ , which has N_{s_x} by N_{s_y} rows.

4-2 3D f-k-k Filter

In section 2-2-3 the 2D f - k filter was introduced. In 3D there are two spatial directions (x, y), i.e. the filter can be extended to a 3D f - k_x - k_y filter.

For this purpose one considers a 3D common receiver gather, $\mathbf{p}(t, x_s, y_s)$, and brings it to the f - k_x - k_y domain by applying a 3-dimensional Fourier transform. Next, a constant frequency slice is selected. This leaves a 2D matrix, which captures the crossline and inline wavenumbers (k_x, k_y) as shown in Figure 4-4a. The minimum wavefield velocity, v_{min} , and the frequency, f , determine the maximum wavenumber, k_{max} , according to equation 2-7;

$$k_{max} = \frac{f}{v_{min}}. \quad (4-1)$$

The total wavenumber, k_T , must be smaller than the maximum wavenumber, k_{max} ,

$$k_T = \sqrt{k_x^2 + k_y^2} < k_{max}. \quad (4-2)$$

Hence the signal "cone" is defined by a circle (see Figure 4-4b). This is repeated for each frequency component, such that the overall f - k_x - k_y mask is a 3D cone (see Figure 4-4c). The cone can be sorted in a 2D view according to section 4-1 as illustrated in Figure 4-4d and Figure 4-4e. Finally, this mask is computed for each receiver gather.

For comparison, a 2D f - k filter is designed for 3D data and plotted in a 2D view in Figure 4-5. Note that the 3D f - k_x - k_y filter (see Figure 4-4) removes significantly more incoherent energy than the 2D f - k filter.

4-3 Results

The 3D f - k_x - k_y filter removes incoherent energy in the crossline and inline direction. The 3D blended acquisition design suggested in this thesis blends shots within the same crossline. Hence, the underlying question is whether an extension of the 2D f - k_x filter to the inline direction provides significant deblending enhancements.

For this purpose the synthetic data of Figure 4-6a are blended: Within each crossline there are 21 sources, which are blended in 3 experiments. For each experiment 7 randomly selected sources are blended with random time delays, i.e. the blending pattern with mixed incoherency is applied. The maximum allowed firing time delay is set to 400 ms. The incoherency measure of the used blending matrix, Γ , is $\mu = 99\%$.

The blended data are deblended with the 3D deblending algorithm. In one case a 2D f - k_x filter is applied (see Figure 4-6c). In the other case a 3D f - k_x - k_y filter is applied (see Figure 4-6d). It is clearly visible that the deblending quality increases significantly with the 3D f - k_x - k_y filter. Figure 4-7 displays a 420 ms time slice of each subplot in Figure 4-6.

In order to quantify the quality gap between the results with 2D and 3D filters, the data of Figure 4-6a are blended with maximum firing time delays varying between 40 ms and 400 ms. The blended data are deblended in one case with a 2D f - k filter, and in the other case with a 3D f - k_x - k_y filter. The resulting quality factors are shown in Figure 4-8.

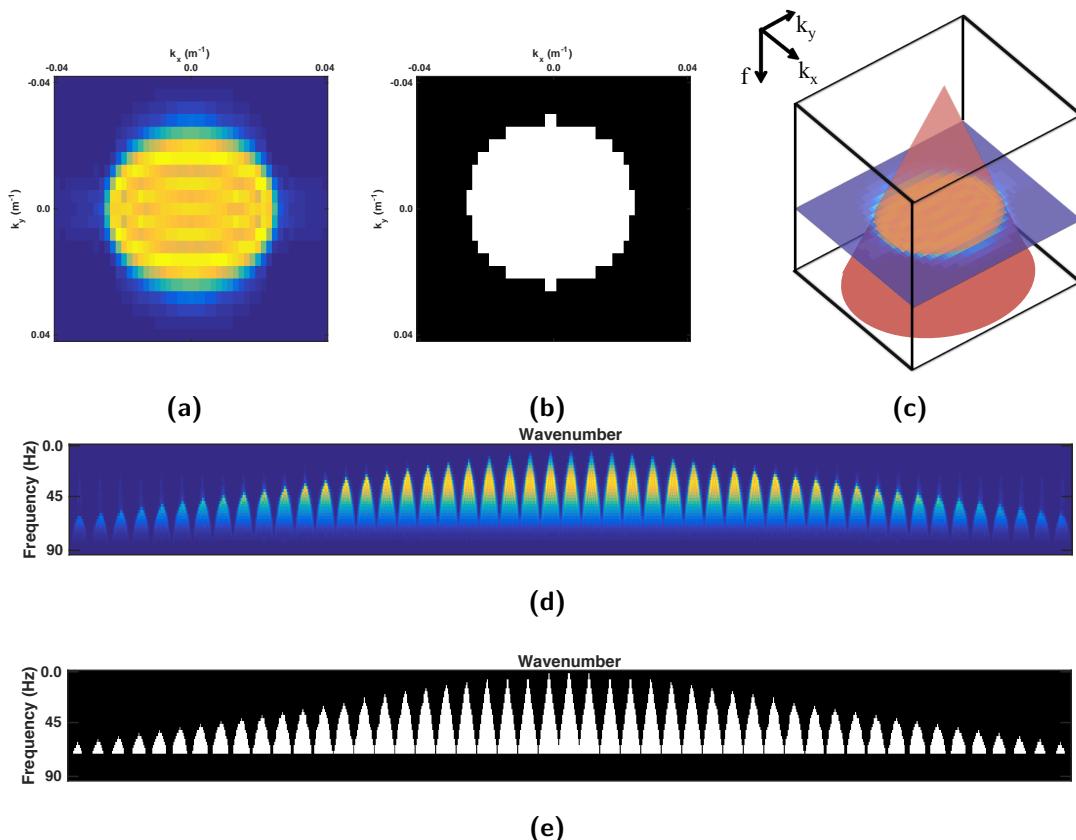


Figure 4-4: Illustration of the 3D f - k_x - k_y -filter. (a) is a 40 Hz frequency slice of the f - k_x - k_y spectrum of the data in Figure 4-2. k_x and k_y refer to the crossline and inline wavenumber respectively. (b) is a 40 Hz frequency slice of the f - k_x - k_y mask, where the white area equals 1 and the black area is 0. (c) shows the 40 Hz frequency slice of (a) sorted in a 3D cube. The red cone represents the edge of the 3D f - k_x - k_y filter mask. (d) and (e) display the f - k_x - k_y data spectrum and mask sorted according to section 4-1, i.e. each sub-cone refers to one inline wavenumber. Note that due to the sorting the wavenumber axis is a mix of crossline and inline wavenumbers. For this reason the wavenumber axis has no labels.

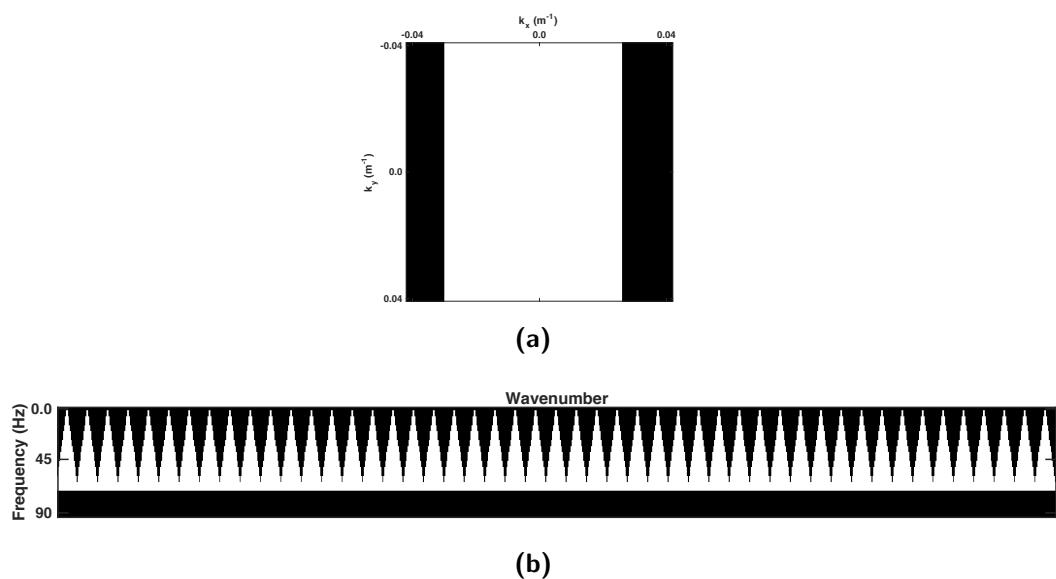


Figure 4-5: 2D f - k_x -filter for 3D data. (a) shows a 40 Hz frequency slice of the f - k_x - k_y spectrum, where the white area equals 1 and the black area is 0. Note that the filter is not affecting the inline wavenumbers k_y . (b) illustrates the 2D f - k_x -filter sorted according to section 4-1. Each cone represents a 2D f - k -filter for a single inline wavenumber.

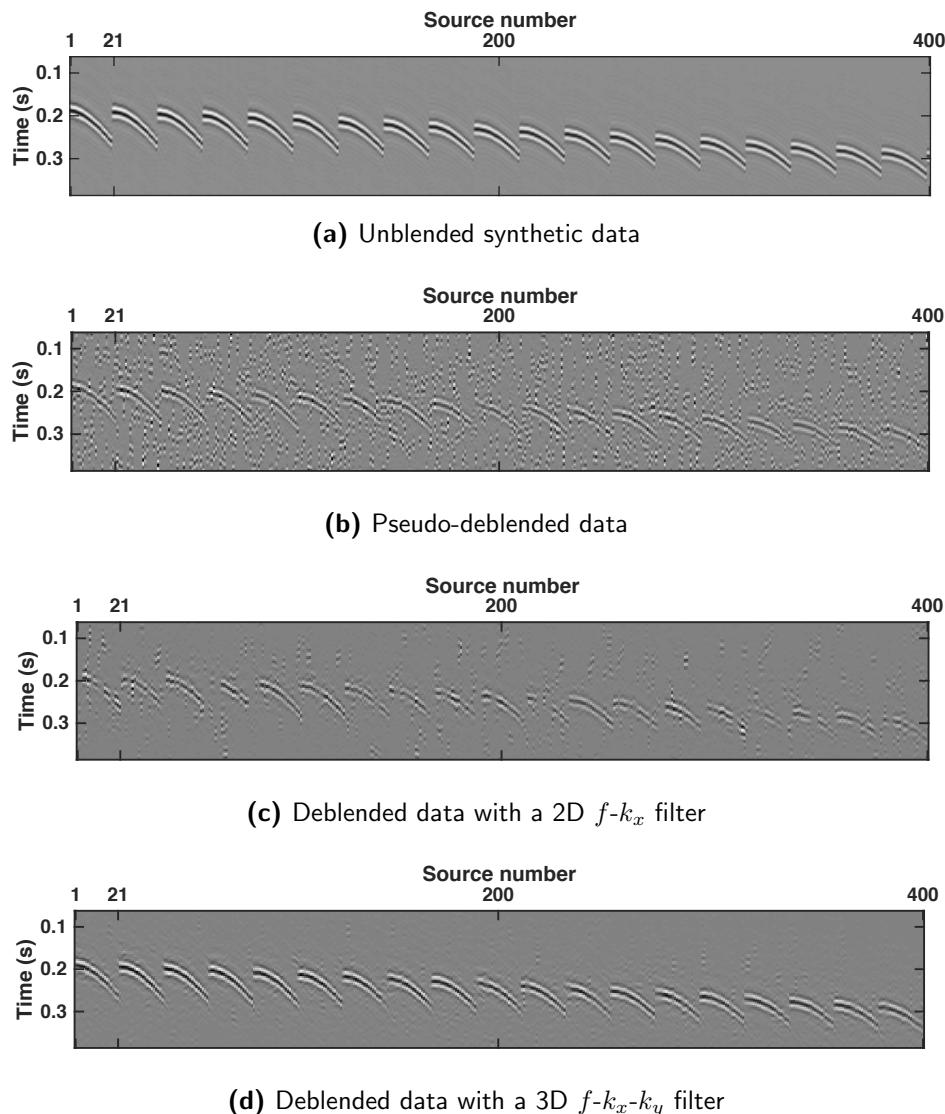


Figure 4-6: (a) is a synthetic 3D common receiver gather in 2D view. The data are generated by 21 crossline shots and 51 inline shots. The shown section is a zoom on the strongest events. Figure 3-4a shows one crossline of the data for all times. The data are blended with a mixed incoherency blending pattern. Then, the 3D deblending algorithm is applied. (b) shows the pseudo-deblended data. In case (c) the algorithm uses a 2D $f-k_x$ filter. In case (d) it uses a 3D $f-k_x-k_y$ filter.

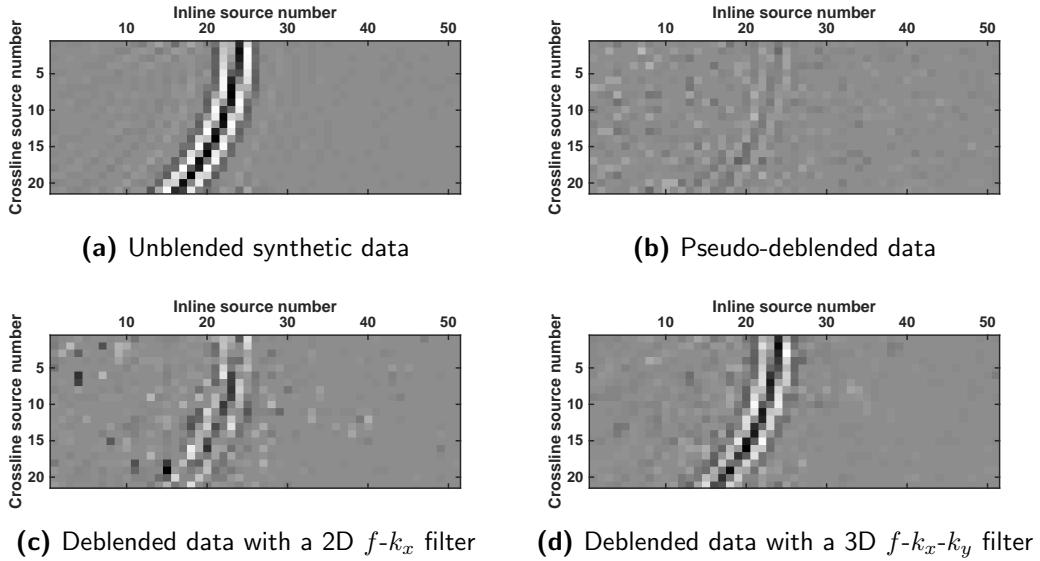


Figure 4-7: (a) - (d) show 420 ms time slices of the data in Figure 4-6.

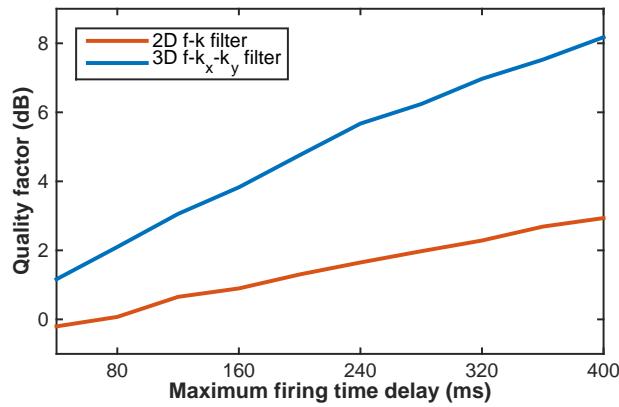


Figure 4-8: Comparison of the deblending quality with a 2D $f-k_x$ filter and a 3D $f-k_x-k_y$ filter. The data in Figure 4-6a are blended with varying maximum firing time delay. Then the blended data are deblended using a 2D $f-k_x$ filter and a 3D $f-k_x-k_y$ filter.

4-4 Conclusions

The presented deblending method uses a coherency constraint. It has been demonstrated that for 3D deblending the quality of the deblended data can be enhanced significantly by extending the coherency constraint to crossline and inline direction.

Chapter 5

Complex Data Results

Chapter 6

Discussion and Conclusions

Bibliography

- Berkhout, A. (1982). *Seismic Migration: Imaging of Acoustic Energy by Wave Field Extrapolation, Part A: Theoretical Aspects*. Elsevier.
- Ibrahim, A. and Sacchi, M. D. (2015). Fast simultaneous seismic source separation using stolt migration and demigration operators. *researchgate.net*.
- Mahdad, A., Doulgeris, P., and Blacquiere, G. (2011). Separation of blended data by iterative estimation and subtraction of blending interference noise. *Geophysics*, 76(3):Q9–Q17.
- van Dedem, E. J. (2002). *3D surface-related multiple prediction*. PhD thesis, TU Delft.
- van Groenestijn, G. J. A. (2010). *Estimation of Primaries and Multiples by Sparse Inversion*. PhD thesis, TU Delft.
- Wu, S. (2014). Seismic deblending by shot repetition. Master's thesis, TU Delft.

Appendix A

The back of the thesis

A-1 An appendix section

A-1-1 An appendix subsection with C++ Listing

```
//  
// C++ Listing Test  
//  
#include <stdio.h>  
for( int i=0;i<10;i++)  
{  
    cout << "Ok\n";  
}
```

1
6

A-1-2 A Matlab Listing

```
%  
% Comment  
%  
n=10;  
for i=1:n  
    disp('Ok');  
end
```

5

Appendix B

Yet another appendix

B-1 Another test section

Ok, all is well.

