

IMPROVED METHODS OF REFLECTION SEISMIC DATA  
PROCESSING FOR VELOCITY ESTIMATION,  
IMAGING, AND INTERPRETATION

by  
Simon S. Luo

UMI Number: 3668324

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3668324

Published by ProQuest LLC (2014). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 - 1346

© Copyright by Simon S. Luo, 2014

All Rights Reserved

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Geophysics).

Golden, Colorado

Date \_\_\_\_\_

Signed: \_\_\_\_\_

Simon S. Luo

Signed: \_\_\_\_\_

Dr. Dave Hale  
Thesis Advisor

Golden, Colorado

Date \_\_\_\_\_

Signed: \_\_\_\_\_

Dr. Terence K. Young  
Professor and Head  
Department of Geophysics

## ABSTRACT

Seismic images and the geologic information they provide contribute significantly to our understanding of the earth's subsurface. In this thesis, I focus on methods relevant for constructing and interpreting seismic images, including methods for velocity estimation, seismic imaging, and interpretation, which together address key aspects of reflection seismic data processing. Specifically, I propose improved methods for semblance-based normal-moveout velocity analysis, for seismic imaging by least-squares migration, and for the automatic extraction of geologic horizons.

To compute a seismic image, an estimate of the subsurface velocity is needed. One common method for constructing an initial velocity model is semblance-based normal-moveout (NMO) velocity analysis, in which semblance spectra are analyzed to identify peaks in semblance corresponding to effective NMO velocities. The accuracy of NMO velocities obtained from semblance spectra depends on the sensitivity of semblance to changes in velocity. By introducing a weighting function in the semblance calculation, I emphasize terms that are more sensitive to velocity changes, which, as a result, increases the resolution of semblance spectra and allows for more accurate NMO velocity estimates.

Following velocity analysis, a seismic image of the subsurface is computed by migrating the recorded data. However, while velocity analysis is an important step in processing reflection seismic data, in practice we expect errors in the velocity models we compute, and these errors can degrade a seismic image. Instead of minimizing the difference between predicted and observed seismic data as is done for conventional migration, I propose to minimize the difference between predicted and time-shifted observed data, where the time shifts are the traveltimes differences between predicted and observed data. With this misfit function, an image computed for an erroneous velocity model contains features similar to those obtained using a more accurate velocity.

Once a seismic image is computed, a common task in interpreting the image is the identification of geologic horizons. As an alternative to manual picking or autotracking, I propose methods to automatically and simultaneously extract all horizons within an image. To extract geologic horizons, a seismic image is unfaulted and unfolded to restore horizons to an undeformed, horizontal state from which they can be easily identified and extracted.

## TABLE OF CONTENTS

ABSTRACT . . . . .	iii
LIST OF FIGURES . . . . .	viii
ACKNOWLEDGMENTS . . . . .	xii
CHAPTER 1 INTRODUCTION . . . . .	1
1.1 Velocity estimation . . . . .	1
1.2 Seismic imaging . . . . .	2
1.3 Interpretation . . . . .	4
1.4 Publications and proceedings . . . . .	5
CHAPTER 2 VELOCITY ANALYSIS USING WEIGHTED SEMBLANCE . . . . .	7
2.1 Summary . . . . .	7
2.2 Introduction . . . . .	7
2.3 Semblance methods . . . . .	9
2.3.1 Conventional semblance . . . . .	9
2.3.2 Conventional semblance rewritten . . . . .	10
2.3.3 Weighted semblance . . . . .	11
2.3.4 Weighting function . . . . .	12
2.3.5 Increasing resolution . . . . .	14
2.4 Results . . . . .	16
2.4.1 Synthetic data examples . . . . .	16
2.4.2 Field data examples . . . . .	22

2.5	Conclusion . . . . .	28
2.6	Acknowledgements . . . . .	29
CHAPTER 3 LEAST-SQUARES MIGRATION IN THE PRESENCE OF VELOCITY ERRORS . . . . .		30
3.1	Summary . . . . .	30
3.2	Introduction . . . . .	30
3.3	Methods . . . . .	34
3.3.1	Linearized waveform inversion . . . . .	34
3.3.2	Dynamic warping . . . . .	35
3.3.3	Inversion of amplitude errors . . . . .	36
3.4	Results . . . . .	38
3.4.1	Synthetic data example . . . . .	39
3.4.2	Field data example . . . . .	46
3.5	Discussion . . . . .	48
3.6	Conclusion . . . . .	52
3.7	Acknowledgement . . . . .	53
CHAPTER 4 NON-VERTICAL DEFORMATIONS FOR SEISMIC IMAGE FLATTENING . . . . .		54
4.1	Summary . . . . .	54
4.2	Introduction . . . . .	54
4.2.1	Structure tensors . . . . .	55
4.3	Image flattening . . . . .	57
4.3.1	Flattening in 2D . . . . .	58
4.3.2	Flattening in 3D . . . . .	68

4.4	Inverse shift vectors? . . . . .	71
4.5	Conclusion . . . . .	73
4.6	Acknowledgements . . . . .	74
CHAPTER 5 UNFAULTING AND UNFOLDING 3D SEISMIC IMAGES . . . . .		75
5.1	Summary . . . . .	75
5.2	Introduction . . . . .	75
5.3	Image unfaulting . . . . .	77
5.4	Image unfolding . . . . .	86
5.5	Horizon extraction . . . . .	92
5.6	Conclusion . . . . .	94
5.7	Acknowledgments . . . . .	95
CHAPTER 6 SUMMARY AND SUGGESTIONS . . . . .		96
REFERENCES CITED . . . . .		100

## LIST OF FIGURES

Figure 2.1	Synthetic CMP gather (a), plot of $b$ values used in the weighting function (b), conventional semblance (c), and weighted semblance (d). . . . .	17
Figure 2.2	Semblance as a function of trial velocity at $\tau = 3.2$ s, for the semblance spectra shown in Figure 2.1. . . . .	18
Figure 2.3	Synthetic CMP gather (a), plot of $b$ values used in the weighting function (b), conventional semblance (c), and weighted semblance (d). . . . .	19
Figure 2.4	Semblance as a function of trial velocity at $\tau = 3.2$ s, for the semblance spectra shown in Figure 2.3. . . . .	20
Figure 2.5	Synthetic CMP gather (a), plot of $b$ values used in the weighting function (b), conventional semblance (c), and weighted semblance (d). . . . .	21
Figure 2.6	Semblance as a function of trial velocity at $\tau = 3.2$ s, for the semblance spectra shown in Figure 2.5. . . . .	22
Figure 2.7	Root-mean-square error of NMO velocities of semblance peaks for noise-free synthetic CMP gathers (a), and for synthetic CMP gathers with signal-to-noise ratio equal to one (b). Gray and black lines correspond to conventional and weighted semblance, respectively. . . . .	23
Figure 2.8	Shot record #3 from <i>Seismic Data Analysis</i> (a), plot of $b$ values used in the weighting function (b), conventional semblance (c), and weighted semblance (d). . . . .	24
Figure 2.9	Shot record #8 from <i>Seismic Data Analysis</i> (a), plot of $b$ values used in the weighting function (b), conventional semblance (c), and weighted semblance (d). . . . .	25
Figure 2.10	Shot record #16 from <i>Seismic Data Analysis</i> (a), plot of $b$ values used in the weighting function (b), conventional semblance (c), and weighted semblance (d). . . . .	26
Figure 2.11	Shot record #30 from <i>Seismic Data Analysis</i> (a), plot of $b$ values used in the weighting function (b), conventional semblance (c), and weighted semblance (d). . . . .	27

Figure 3.1	A simple example. The (a) predicted data (black) and observed data (red); (b) predicted data (black) and shifted observed data (blue); (c) normalized misfit function computed with the predicted and observed data shown in (a); and (d) normalized misfit function computed with the predicted and shifted observed data shown in (b). A local optimization method beginning at the position indicated by the white circle will converge to a local minimum in (c), but will find the global minimum, indicated by the magenta star, in (d). . . . .	32
Figure 3.2	The (a) true slowness model, (b) true background slowness model, (c) true reflectivity computed as the difference between the true slowness squared and the true background slowness squared, and (d) LSM image. . . . .	39
Figure 3.3	The (a,b) difference between the true background slowness shown in Figure 3.2(b) and the background slowness used for migration; (c) LSM image computed for the background slowness with error shown in (a); (d) LSM image computed for the background slowness with error shown in (b); (e) LSMA image computed for the background slowness with error shown in (a); and (f) LSMA image computed for the background slowness with error shown in (b). . . . .	40
Figure 3.4	Zoomed views of the areas enclosed by yellow boxes in the (a) LSM image in Figure 3.3c, (b) LSMA image in Figure 3.3e, and (c) true reflectivity in Figure 3.2c. . . . .	42
Figure 3.5	Zoomed views of the areas enclosed by green boxes in the (a) LSM image in Figure 3.3d, (b) LSMA image in Figure 3.3f, and (c) true reflectivity in Figure 3.2c. . . . .	43
Figure 3.6	Normalized (a) data and amplitude misfit and (b) model misfit for LSM and LSMA. Here, <i>true background</i> refers to the true background slowness shown in Figure 3.2(b), while <i>incorrect background</i> refers to the background slowness with error shown in Figure 3.3(a). . . . .	45
Figure 3.7	Normalized (a) data and amplitude misfit and (b) model misfit for LSM and LSMA. Here, <i>true background</i> refers to the true background slowness shown in Figure 3.2(b), while <i>incorrect background</i> refers to the background slowness with error shown in Figure 3.3(b). . . . .	45
Figure 3.8	The (a) laterally invariant and (b) optimized slowness models used for migration, and the (c) difference between (b) and (a). . . . .	47
Figure 3.9	The (a) LSM image and (b) LSMA image computed for the laterally invariant slowness model shown in Figure 3.8(a), and the (c) LSM image computed for the optimized slowness model shown in Figure 3.8(b). . . . .	49

Figure 3.10	Zoomed views of the areas enclosed by yellow boxes in (a) Figure 3.9(a), (b) Figure 3.9(b), and (c) Figure 3.9(c). . . . .	50
Figure 3.11	For the shot located at distance 1.85 km, the (a) observed data, (b) predicted data computed using the laterally invariant slowness model shown in Figure 3.8(a) and the LSMA image shown in Figure 3.9(b), and (c) traveltimes shifts between (a) and (b). . . . .	51
Figure 4.1	Normal vectors (a) and linearity (b) computed for a 2D seismic image. . . . .	56
Figure 4.2	A unit area (a) in flattened space maps to a rotated unit area (b) or a sheared unit area (c) in present-day space. . . . .	60
Figure 4.3	A synthetic image (a) flattened by vertical shear (b) with the horizontal (c) and vertical (d) components of the shift vector field $\mathbf{r}(\mathbf{u})$ . . . . .	63
Figure 4.4	A synthetic image (a) flattened by rotation (b) with the horizontal (c) and vertical (d) components of the shift vector field $\mathbf{r}(\mathbf{u})$ . . . . .	64
Figure 4.5	A seismic image (a) flattened by vertical shear (b) with the horizontal (c) and vertical (d) components of the shift vector field $\mathbf{r}(\mathbf{u})$ . . . . .	65
Figure 4.6	A seismic image (a) flattened by rotation (b) with the horizontal (c) and vertical (d) components of the shift vector field $\mathbf{r}(\mathbf{u})$ . . . . .	66
Figure 4.7	A seismic image (a) flattened by rotation (b) with the horizontal (c) and vertical (d) components of the shift vector field $\mathbf{r}(\mathbf{u})$ . . . . .	67
Figure 4.8	A 3D seismic image (a) flattened by rotation (b) with the inline shift $r_1$ (c), crossline shift $r_2$ (d), and vertical shift $r_3$ (e) of the shift vector field $\mathbf{r}(\mathbf{u})$ . . . . .	69
Figure 5.1	A fault dip-separation vector. At $\mathbf{w}_f$ on the footwall side of the fault, the dip separation is zero. At $\mathbf{w}_h$ on the hanging wall side of the fault, the dip-separation vector $\tilde{\mathbf{t}}(\mathbf{w}_h)$ specifies the location of the image sample that aligns with the sample on the footwall side. . . . .	78
Figure 5.2	A seismic image (a) overlaid with the fault throw, and the blended neighbor interpolation (b) of the fault throw. . . . .	80
Figure 5.3	A seismic image (a) overlaid with the fault throw, and the unfaulted image (b). . . . .	81
Figure 5.4	The seismic image shown in Figure 5.3(a) is unfaulted and unfolded (a) using the composite shift vectors (b). . . . .	82

Figure 5.5	The inline (a) and crossline (b) components of the composite shift vectors used to unfault and unfold the seismic image shown in Figure 5.3(a). . . . .	83
Figure 5.6	A seismic image (a) overlaid with the fault throw, and the unfaulted image (b). . . . .	84
Figure 5.7	The seismic image shown in Figure 5.6(a) is unfaulted and unfolded (a) using the composite shift vectors (b). . . . .	85
Figure 5.8	At any point $\mathbf{w}$ on a smooth surface, there exist vectors $D_1\mathbf{w}$ and $D_2\mathbf{w}$ tangent to the surface and orthogonal to a normal vector $\mathbf{n}$ . The mapping $\mathbf{w}(\mathbf{u})$ , with inverse mapping $\mathbf{u}(\mathbf{w})$ , determines local metric properties around points $\mathbf{w}$ and $\mathbf{u}$ . For seismic image flattening, we solve for an approximately isometric mapping $\mathbf{w}(\mathbf{u})$ that flattens geologic horizons. . . . .	88
Figure 5.9	Geologic horizons extracted using the composite shift vector fields shown in Figure 5.4(b) and Figure 5.5(a) and Figure 5.7(b) (b). On each horizon, colors represent the vertical component of the corresponding composite shift vector field. . . . .	93

## ACKNOWLEDGMENTS

Although only one name appears on the cover of this thesis, no degree is earned without help from others, and mine certainly is no exception.

First and foremost, I would like to thank my advisor and mentor, Dave Hale, for his continued guidance and support during my time at the Colorado School of Mines. It has been both a pleasure and a challenge working with and learning from him these past years. I am also grateful to the professors in the Center for Wave Phenomena — Norm Bleistein, Dave Hale, Ken Lerner, Paul Sava, Roel Snieder, and Ilya Tsvankin — for creating a wonderful environment for research and learning, and for their willingness to share their expertise and educate students. Thanks also to the sponsors of CWP for supporting me and other students financially, for donating data to test ideas on, and for providing internship opportunities. Thanks to my committee members, Mahadevan Ganesh, Yaoguo Li, Bruce Trudgill, Terry Young, and Andrzej Szymczak for helpful suggestions on my research and thesis.

I would also like to thank the faculty and staff in CWP and the Geophysics department at the Colorado School of Mines. Pam Kraus, Michelle Szobody, and Dawn Umpleby were helpful and efficient when dealing with administrative issues, allowing students to focus on their own work. Barbara McLennon and Shingo Ishida did a great job organizing and assisting with research publications, and designing our consortium website and brochures. Diane Witters was the best source of information for anything related to writing, and in addition she also supported and counseled students through stressful times. John Stockwell helped with computer issues, and was always eager to discuss geophysical, mathematical, and computational problems with students.

I would especially like to thank my many teachers over the years, including the professors and instructors whom I have taken courses from, friends and fellow students within CWP who have offered suggestions and feedback, colleagues both in and outside of Mines whom

I have had discussions with, and mentors and collaborators who have guided me during my internships. I am extremely grateful to those that have contributed to my education in any way.

Finally, I would like to thank my parents for their constant support and encouragement, for allowing me the freedom to pursue my interests and find my own path, and for being understanding of the mistakes I have made along the way.

# CHAPTER 1

## INTRODUCTION

The primary goal of exploration seismology is to provide information about the earth's subsurface. Often, this information is obtained from a seismic image and from the geologic knowledge provided by the image. Seismic imaging requires the conversion of seismic data, recorded in time, to an image of the subsurface in depth. Because this conversion requires an estimate of the velocity at which seismic waves propagate in the subsurface, a necessary task in constructing an image of the subsurface is to perform velocity analysis. Following velocity analysis, an image is computed by migrating the recorded seismic data in order to map the data to subsurface reflectors from which they originated. Finally, once the image is computed, it must be interpreted to identify geologic features of interest, such as geologic horizons. In this thesis, I focus on the three aspects of reflection seismic data processing mentioned above: velocity estimation, seismic imaging or seismic migration, and interpretation.

### 1.1 Velocity estimation

To compute a seismic image, it is necessary to first estimate the subsurface properties that affect the kinematics of wave propagation. For example, in the simplest case in which we assume an acoustic earth, the property that we must estimate is the P-wave velocity. A common approach for conventional velocity analysis consists of two steps (Hill & Rüger, 2008; Toldi, 1985): (1) stacking or normal-moveout (NMO) velocity analysis, and (2) conversion of NMO velocities to interval velocities. While this approach is most appropriate for simple subsurface models that are laterally invariant, it can also be used to construct starting velocity models for more sophisticated methods of velocity analysis such as full-waveform inversion (Lailly, 1983; Pratt *et al.*, 1998; Tarantola, 1984), which seeks the velocity model that best predicts the recorded seismic data, or migration velocity analysis (Al-Yahya, 1989; Biondi & Sava, 1999; Fowler, 1985; Sava & Biondi, 2004; Shen, 2004; Symes & Carazzone,

1991), which constructs an optimal velocity model by analyzing common image gathers. One difficulty with migration velocity analysis and full-waveform inversion is that the associated inverse problems can be highly nonlinear, and local minima in the objective functions can result in unphysical and inaccurate models. In such cases, an improved starting model can help to avoid local minima.

The focus of **Chapter 2** is a method for improving the resolution of semblance-based NMO velocity analysis (Neidell & Taner, 1971; Taner & Koehler, 1969). Semblance is a coherence measure that has similarities to normalized crosscorrelation and also to summation, but, for the purpose of NMO velocity analysis, semblance has the greatest discriminating power (Neidell & Taner, 1971). Because semblance spectra often are computed and used to pick NMO velocities, higher resolution spectra are desirable because they can directly improve one's ability to identify, distinguish, and pick peaks in semblance corresponding to optimal effective NMO velocities. To increase the resolution of semblance spectra, I introduce a weighting function to minimize semblance, and, somewhat surprisingly, this weighting function results in an increase in resolution. I demonstrate on synthetic and field data that the use of this weighting function increases the resolution of semblance spectra, and I also demonstrate on synthetic data that, along with the increase in resolution, the weighting function also results in an increase in accuracy compared to conventional semblance.

## 1.2 Seismic imaging

Following velocity analysis, a subsequent step in processing reflection seismic data is to construct a seismic image by migrating the recorded data. Methods for migration require the solution of an appropriate wave equation, which are commonly obtained either by constructing a high-frequency asymptotic solution or by finite differencing. High-frequency, i.e., ray-based, methods are attractive because they are computationally efficient; a drawback, however, is that the high-frequency assumption becomes less accurate in the presence of complex subsurface structures. As a result, ray-based migration images tend to degrade with increasing structural complexity.

Finite-difference solutions of the wave equation are more accurate than ray-based solutions, and finite-difference migrations can offer improved images compared to ray-based migrations. Finite-difference migrations typically can be categorized as either one-way or two-way migrations. One-way migrations are based on solutions of a one-way paraxial approximation to the wave equation, while two-way migrations are based on solutions of a full, i.e., unapproximated, wave equation. Compared to ray-based migrations, both one-way and two-way migrations yield improved images when the subsurface produces complex wave propagation phenomena such as multipathing. In addition, compared to one-way migrations, two-way migrations produce more accurate reflector amplitudes and improved images of steeply dipping reflectors (Mulder & Plessix, 2004).

In **Chapter 3**, I propose a method for least-squares migration for velocity models that contain errors. Least-squares migration (Dai, 2012; Köhl & Sacchi, 2003; Nemeth *et al.*, 1999; Østmo & Plessix, 2002; Plessix & Mulder, 2004) is a two-way migration that seeks to invert for the seismic image by minimizing a data-domain objective function. Compared to other methods such as reverse-time migration (Baysal *et al.*, 1983; Levin, 1984; Loewenthal & Mufti, 1983; McMechan, 1983; Whitmore, 1983) that compute images using the adjoint of a linearized forward modeling operator, least-squares migration images offer improved resolution and better balanced amplitudes by using the pseudoinverse of the forward operator.

Least-squares migration, like other methods for migration, relies on an accurate velocity model to compute an accurate image. While velocity analysis is a critical step in reflection seismic data processing, in practice we expect errors in the velocity models we compute. Such errors can result from inadequacies in the methods used for velocity analysis, inaccuracies in our assumptions about the earth’s subsurface (e.g., the assumption of an acoustic and isotropic earth), and insufficiencies in the data we record for constraining the relevant subsurface properties. Because errors in velocity are unavoidable in practice, it is worthwhile to consider the case of migration when the velocity model is erroneous.

The method for least-squares migration proposed in Chapter 3 introduces a time-shift operator into the objective function that is minimized to compute a seismic image. This time-shift operator shifts the observed seismic data to reduce traveltime differences between predicted and observed data that can result, for example, from errors in velocity, but more generally result from an inability to exactly model seismic wave propagation in the subsurface. In any case, differences in traveltime between predicted and observed data lead to degraded migration images. Thus, by introducing a time-shift operator to correct for traveltime differences, I improve the migration images computed when using an inaccurate velocity model.

### 1.3 Interpretation

Following the computation of a seismic image of the subsurface, we must interpret the image to identify geologic features of interest. These features typically are structural or stratigraphic features capable of forming hydrocarbon traps, and such features are commonly identified by picking, either manually or perhaps while aided by a computer, geologic horizons. However, manual or computer-aided interpretation of geologic horizons can be time-consuming and impractical, especially for large 3D images. Thus, automatic methods for interpretation are preferable.

In **Chapter 4** and **Chapter 5**, I propose methods for the automatic and simultaneous extraction of all geologic horizons within a seismic image. Each geologic horizon corresponds to an isochron surface — a surface of constant geologic time. Due to geologic deformation, however, surfaces of constant geologic time are rarely aligned with axes of seismic images; hence the need for interpretation or extraction of geologic horizons.

In Chapter 4, I propose a method for unfolding seismic images to restore geologic horizons to an undeformed, horizontal state. In an unfolded image, surfaces of constant geologic time are aligned with the vertical axis of the image, making the identification and extraction of geologic horizons simple. This unfolding process is commonly referred to as *seismic image flattening*, and most methods for seismic image flattening (Lomask *et al.*, 2006; Parks, 2010;

Stark, 2004) are limited to the use of vertical shifts only. Because of this limitation, these methods have difficulty flattening seismic images that contain geologic features resulting from non-vertical deformations without significantly distorting such features. The flattening method that I propose in Chapter 4 uses non-vertical vector shifts to flatten a seismic image, which results in less distortion of geologic features while flattening.

Another common limitation of methods for image unfolding or image flattening, including the method proposed in Chapter 4, is their inability to properly handle geologic faults. Methods for image unfolding require an estimate of the local orientation, e.g., the dip or slope, of image features in order to flatten an image. The difficulty with geologic faults is twofold: first, because local slopes are discontinuous across a fault, methods that estimate slopes by averaging image features within a window may yield inaccurate slopes; and second, because the slope at a fault is undefined, any slopes estimated at fault locations will be inherently flawed.

I address this limitation in Chapter 5, in which I propose a method for unfauling and unfolding seismic images. First, a seismic image is unfaled by shifting image features according to dip-separation vectors computed at fault locations (Hale, 2013b) and then interpolated at locations between faults. Then, the unfaled image is subsequently unfolded by flattening the image. The proposed method for image flattening is similar to that discussed in Chapter 4, as it flattens an image using non-vertical shifts, but it differs in the form of the partial-differential equations that are solved. To minimize distortions of geologic features, the method proposed in Chapter 5 seeks to flatten an image while preserving metric properties such as angle, length, and volume in the flattened image. Unfauling and unfolding a seismic image allows for the automatic simultaneous extraction of all geologic horizons, including those interrupted by geologic faults.

## **1.4 Publications and proceedings**

Over the course of my degree program, I have contributed several publications and conference abstracts. Chapters 2, 3, and 5 of this thesis have been published in whole or in part

in the journal *Geophysics*:

- **Luo, S.**, and D. Hale, 2011, Velocity analysis using weighted semblance: *Geophysics*, 77(2), U15-U22.
- **Luo, S.**, and D. Hale, 2013, Unfaulting and unfolding 3D seismic images: *Geophysics*, 78(4), O45-O56.
- **Luo, S.**, and D. Hale, 2014, Least-squares migration in the presence of velocity errors: *Geophysics*, 79(4), S153-S161.

The work discussed in Chapter 4 was one of several expanded abstracts presented at annual meetings of the Society of Exploration Geophysicists:

- **Luo, S.**, and D. Hale, 2010, Velocity analysis using weighted semblance: 80th Annual Meeting of the Society of Exploration Geophysicists, Expanded Abstracts.
- **Luo, S.**, and D. Hale, 2011, Non-vertical deformations for seismic image flattening: 81st Annual Meeting of the Society of Exploration Geophysicists, Expanded Abstracts.
- **Luo, S.**, and P. Sava, 2011, A deconvolution-based objective function for wave-equation inversion: 81st Annual Meeting of the Society of Exploration Geophysicists, Expanded Abstracts.
- **Luo, S.**, and D. Hale, 2012, Unfaulting and unfolding 3D seismic images: 82nd Annual Meeting of the Society of Exploration Geophysicists, Expanded Abstracts.
- **Luo, S.**, and D. Hale, 2013, Separating traveltimes and amplitudes in waveform inversion: 83rd Annual Meeting of the Society of Exploration Geophysicists, Expanded Abstracts.
- **Luo, S.**, and D. Hale, 2014, Least-squares migration in the presence of velocity errors: 84th Annual Meeting of the Society of Exploration Geophysicists, Expanded Abstracts.

## CHAPTER 2

### VELOCITY ANALYSIS USING WEIGHTED SEMBLANCE

Modified from a paper published in *Geophysics*, 2011, 77 (2), U15-U22

Simon Luo<sup>1</sup> and Dave Hale<sup>1</sup>

#### 2.1 Summary

Increasing the resolution of semblance-based velocity spectra, or semblance spectra, is useful for estimating normal moveout velocities, as increased resolution can help to distinguish peaks in the spectra. The resolution of semblance spectra depends on the sensitivity of semblance to changes in velocity. By weighting terms in the semblance calculation that are more sensitive to changes in velocity, we can increase resolution. Our implementation of weighted semblance is a straightforward extension of conventional semblance. Somewhat surprisingly, we increase resolution by choosing an offset-dependent weighting function that *minimizes* semblance. We test our method on both synthetic and field data, and our tests confirm that weighted semblance provides higher resolution than conventional semblance.

#### 2.2 Introduction

Normal moveout (NMO) velocity analysis using semblance spectra (Taner & Koehler, 1969) is an important first step toward building a velocity model. NMO velocity analysis requires picking peaks in semblance spectra, and the resolution of these spectra affects one's ability to distinguish and pick individual peaks. For example, in cases where there are interfering events such as multiples in a common midpoint (CMP) gather, it may be difficult to differentiate the semblance peaks corresponding to the primary events from those corresponding to the interfering events. Higher resolution would better distinguish the different sets of semblance peaks in this situation.

---

<sup>1</sup> Center for Wave Phenomena, Department of Geophysics, Colorado School of Mines

Semblance is a normalized, squared correlation of NMO-corrected seismic data with a constant. Correlation with a constant implies an assumption that there is no amplitude or phase variation with offset (Corcoran & Seriff, 1993; Sarkar *et al.*, 2000). When this assumption is violated, semblance may no longer be an accurate measure of NMO velocity. To address this problem, modified semblance coefficients that measure correlation with a trend (Fomel, 2009; Sarkar *et al.*, 2002; Yan & Tsvankin, 2008) rather than a constant have been proposed.

Correlation with a trend is equivalent to weighted correlation with a constant, and it has been shown that weighting terms in a correlation coefficient calculation that are sensitive to changes in velocity can increase the resolution of the corresponding velocity spectra. For example, Celis & Larner (2002) introduce a selective-correlation sum that improves the resolution of velocity spectra by discarding crosscorrelations between traces with relatively small differential moveout of events. Selective-correlation is effectively a weighted crosscorrelation sum with weights of either zero or unity, depending on the differential moveout between traces.

We can likewise increase the resolution of semblance spectra by weighting terms in the conventional semblance calculation. Unlike Celis & Larner (2002), however, we do not discard terms in the semblance calculation but instead weight all terms on the basis of their sensitivity to changes in velocity. Our implementation of *weighted semblance* is based in part on work presented by Hale (2009b). Hale defines a weighted semblance coefficient that conforms to structural features apparent in seismic images. We define a different semblance coefficient by choosing a different weighting scheme.

In this paper we describe a method to compute weighted semblance for the purpose of increasing the resolution of semblance spectra. To increase resolution, we use an offset-dependent weighting function to minimize semblance, while maintaining a normalized semblance value between zero and one. We test the method on both synthetic and field data to compare the resolution of weighted and conventional semblance. The method is easy to

implement, and its computational cost is comparable to that of conventional semblance.

## 2.3 Semblance methods

Weighted semblance is a straightforward extension of conventional semblance. In this section, we first discuss conventional semblance, and then introduce our implementation of weighted semblance. We then derive the weighting function and show how it is used to increase resolution.

### 2.3.1 Conventional semblance

Conventional semblance is a normalized coherency measure that was first defined by Taner & Koehler (1969). A comparison of semblance and other coherency measures can be found in Neidell & Taner (1971). Semblance is routinely used to estimate NMO velocity as a function of zero-offset time. Following normal moveout correction of a CMP gather, semblance as defined by Neidell & Taner (1971) is computed as

$$s[i] = \frac{\sum_{j=i-M}^{i+M} \left( \sum_{k=0}^{N-1} q[j, k] \right)^2}{N \sum_{j=i-M}^{i+M} \sum_{k=0}^{N-1} q[j, k]^2}, \quad (2.1)$$

where  $i$  and  $j$  are time sample indices,  $k$  is a trace number, and  $q[j, k]$  is the trace amplitude at time index  $j$  and trace number  $k$  of the NMO-corrected gather. The inner sums over  $k$  correspond to  $N$  NMO-corrected traces in a CMP gather, while the outer sums correspond to a time-smoothing window with length  $2M + 1$  centered at time index  $i$ . Here, the time-smoothing is performed by a boxcar filter.

In general, we are free to use any time-smoothing filter, but in practice, it is often a good idea to replace a boxcar filter with one that decays more smoothly. For the examples shown in this paper, the boxcar filter is replaced with a two-sided decaying exponential filter, which we represent by an additional weighting function  $h[j]$ . The derivations are independent of

the choice of  $h[j]$ , so its exact form is not important. We write conventional semblance as

$$s_c[i] = \frac{\sum_j h[i-j] \left( \sum_k q[j,k] \right)^2}{N \sum_j h[i-j] \sum_k q[j,k]^2}, \quad (2.2)$$

where it is assumed that the unspecified summation limits include all indices for which the summation terms are defined.

The semblance value reflects how well the moveout path corresponding to the trial NMO velocity fits the moveout of signal in the data. A good fit produces a peak in the semblance spectrum, whereas a poor fit produces semblance values closer to zero. Assuming there is no noise and no signal amplitude variation with offset, semblance is maximized when the values of  $q[j,k]$  do not vary with index  $k$ . That is,  $s_c[i] = 1$  when the NMO-corrected events are aligned across traces at a single time index  $i$ .

The resolution of semblance spectra depends on the sensitivity of NMO times to changes in velocity. If a small change in trial velocity results in a relatively large change in NMO time, the semblance value will change rapidly with the mismatch between the NMO times corresponding to the trial velocity and the correct velocity. The greater the change in NMO time for a change in trial velocity, the higher the resolution of the semblance spectrum.

### 2.3.2 Conventional semblance rewritten

Before we consider weighted semblance, let us introduce an alternative expression for conventional semblance. We express conventional semblance as a normalized correlation coefficient by first defining a reference trace  $r[j]$  as a summation over trace number (equivalently, a stack over offset) of the NMO-corrected traces in the CMP gather:

$$r[j] \equiv \sum_k q[j,k]. \quad (2.3)$$

To simplify notation, we also define

$$\begin{aligned}
C_{rq}[i] &\equiv \sum_j h[i-j] \sum_k r[j]q[j,k], \\
C_{rr}[i] &\equiv \sum_j h[i-j] \sum_k r[j]^2, \\
C_{qq}[i] &\equiv \sum_j h[i-j] \sum_k q[j,k]^2.
\end{aligned} \tag{2.4}$$

Conventional semblance  $s_c[i]$  can then be written as

$$s_c[i] = \frac{C_{rq}[i]^2}{C_{rr}[i]C_{qq}[i]}. \tag{2.5}$$

Equation 2.5 and equation 2.2 are equivalent expressions for conventional semblance.

### 2.3.3 Weighted semblance

To obtain weighted semblance, we modify conventional semblance by introducing weights  $w[j, k]$  into equations 2.4:

$$\begin{aligned}
W_{rq}[i] &\equiv \sum_j h[i-j] \sum_k w[j,k]r[j]q[j,k], \\
W_{rr}[i] &\equiv \sum_j h[i-j] \sum_k w[j,k]r[j]^2, \\
W_{qq}[i] &\equiv \sum_j h[i-j] \sum_k w[j,k]q[j,k]^2.
\end{aligned} \tag{2.6}$$

Then, weighted semblance  $s_w[i]$  is given by

$$s_w[i] = \frac{W_{rq}[i]^2}{W_{rr}[i]W_{qq}[i]}. \tag{2.7}$$

Note the similarity between equation 2.5 and equation 2.7, and note that weighted semblance is equal to conventional semblance for  $w[j, k] = 1$ . Moreover, it can be shown using the Cauchy-Schwarz inequality that weighted semblance is bounded between zero and one if the weights  $w[j, k]$  and  $h[j]$  are non-negative.

### 2.3.4 Weighting function

We choose a weighting function  $w[j, k]$  to emphasize terms in the semblance calculation that are most sensitive to changes in velocity.

The form of the weighting function reflects the change in NMO time for a given change in velocity. Consider the first-order Taylor series expansion of the hyperbolic moveout equation about the correct slowness-squared  $\tilde{\gamma}$ :

$$t[j, k] = \sqrt{\tau[j]^2 + \tilde{\gamma}x[k]^2} + \frac{x[k]^2}{2\sqrt{\tau[j]^2 + \tilde{\gamma}x[k]^2}} (\gamma - \tilde{\gamma}), \quad (2.8)$$

where  $\tau[j]$  is the zero-offset time at time index  $j$ ,  $x[k]$  is the offset at trace number  $k$ , and  $\gamma \equiv 1/v^2$  where  $v$  is the velocity. A tilde indicates the correct value—i.e.,  $\tilde{\gamma} \equiv 1/\tilde{v}^2$  where  $\tilde{v}$  is the correct velocity. The correct NMO time is given by  $\tilde{t}[j, k] = \sqrt{\tau[j]^2 + \tilde{\gamma}x[k]^2}$ , and we can rearrange equation 2.8 as

$$t[j, k] - \tilde{t}[j, k] = \frac{x[k]^2}{2\tilde{t}[j, k]} (\gamma - \tilde{\gamma}). \quad (2.9)$$

Thus, the change in NMO time that results from a small change in velocity is proportional to offset squared and inversely proportional to time.

To reflect this proportionality, we choose a weighting function  $w[j, k]$  that has a similar dependency on offset and time:

$$w[j, k] = a + b \frac{c[j]x[k]^2}{t[j, k]}, \quad (2.10)$$

where  $a$  and  $b$  are parameters to be determined, and  $c[j]$  is calculated as the ratio of the zero-offset time to the average offset squared:

$$c[j] = \frac{\tau[j]N}{\sum_k x[k]^2}. \quad (2.11)$$

Multiplying by  $c[j]$  ensures that  $b$  is unitless.

The relative values of the parameters  $a$  and  $b$  in equation 2.10 effectively determine how the far offsets are weighted. In cases where we expect large weights for the farthest offsets,

the ratio of  $b$  to  $a$  must approach infinity. To satisfy this condition more easily, we choose

$$a = 1 - b, \quad (2.12)$$

so that

$$w[j, k] = 1 - b + b \frac{c[j]x[k]^2}{t[j, k]}. \quad (2.13)$$

In addition, we only allow  $b$  values between zero and one. Bounding  $b$  ensures that the weighting function is non-negative, which is a sufficient condition for weighted semblance to remain normalized between zero and one.

After substituting equation 2.13 for  $w[j, k]$  into equations 2.6, we have the weighted semblance

$$s_w[i] = \frac{W_{rq}[i]^2}{W_{rr}[i]W_{qq}[i]}, \quad (2.14)$$

where

$$\begin{aligned} W_{rq}[i] &= (1 - b)C_{rq}[i] + bB_{rq}[i], \\ W_{rr}[i] &= (1 - b)C_{rr}[i] + bB_{rr}[i], \\ W_{qq}[i] &= (1 - b)C_{qq}[i] + bB_{qq}[i], \end{aligned} \quad (2.15)$$

where  $C_{rq}[i]$ ,  $C_{rr}[i]$ , and  $C_{qq}[i]$  are defined in equations 2.4, and  $B_{rq}[i]$ ,  $B_{rr}[i]$ , and  $B_{qq}[i]$  are defined as

$$\begin{aligned} B_{rq}[i] &\equiv \sum_j h[i - j] \sum_k \frac{c[j]x[k]^2}{t[j, k]} r[j]q[j, k], \\ B_{rr}[i] &\equiv \sum_j h[i - j] \sum_k \frac{c[j]x[k]^2}{t[j, k]} r[j]^2, \\ B_{qq}[i] &\equiv \sum_j h[i - j] \sum_k \frac{c[j]x[k]^2}{t[j, k]} q[j, k]^2. \end{aligned} \quad (2.16)$$

Weighted semblance is now a function of the parameter  $b$ . Note that when  $b = 0$ , the weighting function  $w[j, k] = 1$ , and weighted semblance is equivalent to conventional semblance. When  $b = 1$ , weighted semblance reduces to the expression given by equations 2.6 and 2.7,

with weighting function  $w[j, k] = c[j]x[k]^2/t[j, k]$ . We will refer to this case as *fully-weighted semblance*, because the farthest offsets are given the most weight when  $b = 1$ .

Note that although the weighting function is derived from the hyperbolic moveout equation, we do not make any assumptions about how the seismic data are NMO-corrected. Because semblance is calculated after NMO correction, we are free to use any moveout equation, hyperbolic or non-hyperbolic, to correct the data. Our method for increasing resolution works in either case.

### 2.3.5 Increasing resolution

To increase the resolution of semblance spectra, we minimize semblance with respect to  $b$ , with the constraint that  $0 \leq b \leq 1$ . Bounding  $b$  ensures that the weights are non-negative, which guarantees that weighted semblance is between zero and one.

It may seem somewhat counterintuitive that minimizing semblance would increase resolution, but recall that when the trial NMO velocity equals the correct velocity, semblance is calculated along what are assumed to be constant trace amplitudes, i.e., amplitude is independent of trace number. If amplitude  $q[j, k]$  is independent of trace index  $k$ , then  $q[j, k] = r[j]/N$  can be pulled out of the summation over  $k$  in equations 2.4 and equations 2.16. Then, semblance is unity, regardless of the weighting function. Because semblance peaks where  $s_w[i] = 1$  are not affected by the weighting function, we can increase the resolution of semblance spectra by choosing a weighting function that minimizes semblance.

To minimize semblance  $s_w[i]$  for any time index  $i$ , we set the derivative of semblance with respect to  $b$  equal to zero (from here on, we omit the time index  $i$  to simplify equations):

$$\frac{ds_w(b)}{db} = 0. \quad (2.17)$$

Solving the above equation, we find that semblance as a function of  $b$  has two stationary points

$$b_1 = \frac{C_{rq}}{C_{rq} - B_{rq}}, \quad (2.18)$$

and

$$b_2 = \left( 1 + \frac{2C_{rq}B_{rr}B_{qq} - B_{rq}A}{2B_{rq}C_{rr}C_{qq} - C_{rq}A} \right)^{-1}, \quad (2.19)$$

where

$$A = C_{rr}B_{qq} + C_{qq}B_{rr}. \quad (2.20)$$

Because we are minimizing semblance, we are interested only in local minima. We compute the second derivative of semblance with respect to  $b$  at the two stationary points to obtain

$$\frac{d^2 s_w(b_1)}{db^2} = \frac{2(B_{rq} - C_{rq})^4}{(B_{rq}C_{qq} - B_{qq}C_{rq})(B_{rq}C_{rr} - B_{rr}C_{rq})}, \quad (2.21)$$

and

$$\frac{d^2 s_w(b_2)}{db^2} = -\frac{2[B_{rq}(B_{rr}C_{qq} + C_{rr}(B_{qq} - 2C_{qq})) + C_{rq}(B_{rr}C_{qq} + B_{qq}(C_{rr} - 2B_{rr}))]^4}{(B_{rq}C_{qq} - B_{qq}C_{rq})(B_{rq}C_{rr} - B_{rr}C_{rq})(B_{rr}C_{qq} - B_{qq}C_{rr})^4}. \quad (2.22)$$

We see by inspection that equation 2.21 is positive—and therefore  $b_1$  corresponds to a local minima—if

$$(B_{rq}C_{qq} - B_{qq}C_{rq})(B_{rq}C_{rr} - B_{rr}C_{rq}) > 0. \quad (2.23)$$

Similarly, equation 2.22 is positive and  $b_2$  corresponds to a local minima if

$$(B_{rq}C_{qq} - B_{qq}C_{rq})(B_{rq}C_{rr} - B_{rr}C_{rq}) < 0. \quad (2.24)$$

Thus to find the  $b$  value within  $0 \leq b \leq 1$  that minimizes semblance, we first compute either  $b_1$  or  $b_2$  depending on whether inequality 2.23 or 2.24 is satisfied, and then if  $b$  is between zero and one, we compute semblance using equation 2.14. If  $b$  is not between zero and one, we simply choose the minimum value of  $s_w(b = 0)$  and  $s_w(b = 1)$ . In this way, we obtain the minimum value of semblance for  $0 \leq b \leq 1$ .

For each zero-offset time, we also scale all weighted semblance values by a constant factor not less than one. We do this because weighted semblance values obtained by minimizing semblance will otherwise tend to be lower, for all moveout velocities, than conventional semblance values. The scaling factor is a function of zero-offset time, and is defined as the

minimum ratio of conventional semblance to weighted semblance. With this scaling factor, weighted semblance never exceeds conventional semblance. Note that because we need to compute  $C_{rr}$ ,  $C_{rq}$ , and  $C_{qq}$  given in equations 2.4 to compute weighted semblance, we can easily compute conventional semblance for little additional cost from equation 2.5.

## 2.4 Results

To illustrate the action of the weighting function  $w[j, k]$  on the resolution of semblance spectra, we compare weighted semblance to conventional semblance for synthetic CMP gathers and for field shot gathers.

### 2.4.1 Synthetic data examples

For all synthetic data examples, the CMP gathers have a cable length of 3 km, receiver group interval spacing of 50 m, and a Ricker wavelet peak frequency of 25 Hz.

The first CMP gather shown in Figure 2.1(a) consists of a series of synthetic primary reflections with linearly increasing NMO velocity. The velocity increases from 2 km/s at zero-offset time  $\tau = 0$  s, to 3 km/s at  $\tau = 4$  s. Figure 2.1(b) depicts the  $b$  values used in the weighting function  $w[j, k]$  that minimizes semblance. Recall that  $b = 0$  corresponds to conventional semblance, while  $b = 1$  corresponds to fully-weighted semblance, which gives the most weight to the farthest offsets. It is interesting to note that the minimum semblance value is not always given by  $b = 1$ ; in other words, simply weighting the far offsets does not always minimize semblance. Figure 2.1(c) and Figure 2.1(d) show the conventional and weighted semblance spectrum, respectively. Note the spread in spectral amplitude across a range of velocities in the conventional semblance spectra. In comparison, in the weighted semblance spectrum, the spread in spectral amplitude has decreased.

We can directly compare semblance peaks by plotting semblance as a function of trial velocity for a chosen zero-offset time. Figure 2.2 depicts this plot for the first synthetic CMP gather at zero-offset time  $\tau = 3.2$  s. In the figure, we see that minimizing semblance has reduced the semblance values at velocities away from the peak. As a result, the weighted

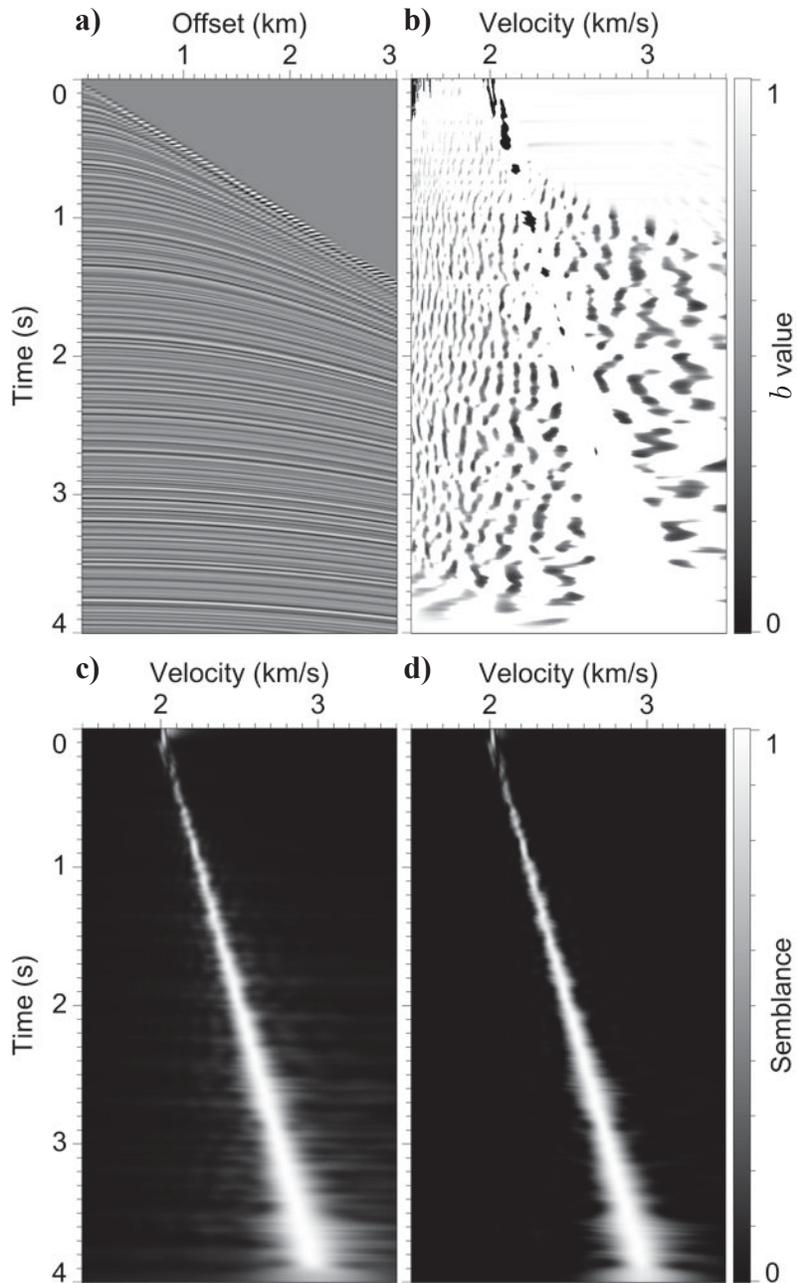


Figure 2.1: Synthetic CMP gather (a), plot of  $b$  values used in the weighting function (b), conventional semblance (c), and weighted semblance (d).

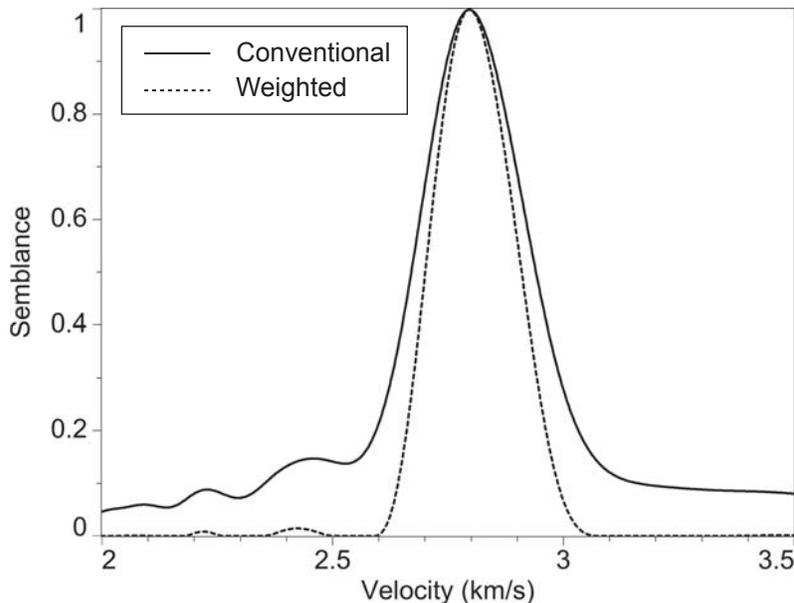


Figure 2.2: Semblance as a function of trial velocity at  $\tau = 3.2$  s, for the semblance spectra shown in Figure 2.1.

semblance peak is sharper than the conventional semblance peak.

Next we add a second set of reflections to the synthetic CMP gather shown in Figure 2.1(a) to simulate interfering multiples. The second set of reflections have NMO velocities that increase linearly from 1.98 km/s at zero-offset time  $\tau = 0$  s, to 2.50 km/s at  $\tau = 4$  s. Figure 2.3(a) depicts the CMP gather, and Figure 2.3(b) depicts a plot of the  $b$  values used in the weighting function  $w[j, k]$ . Figure 2.3(c) and Figure 2.3(d) depict the conventional and weighted semblance spectrum, respectively. Notice that with the second set of reflections, the weighted semblance spectrum now has a smaller range of spectral amplitudes (i.e., smaller peak semblance value) compared to the conventional semblance spectrum. This is a result of minimizing semblance. A necessary assumption for this minimization was that the NMO-corrected trace amplitudes are constant for the correct trial velocity. For these synthetic data, and for field data especially, this assumption is not correct. Thus, in minimizing semblance, we expect in some cases for the peak amplitudes to decrease. Nevertheless, as confirmed by the semblance curve in Figure 2.4, the weighted semblance spectrum affords higher resolution as it better distinguishes the two sets of semblance peaks.

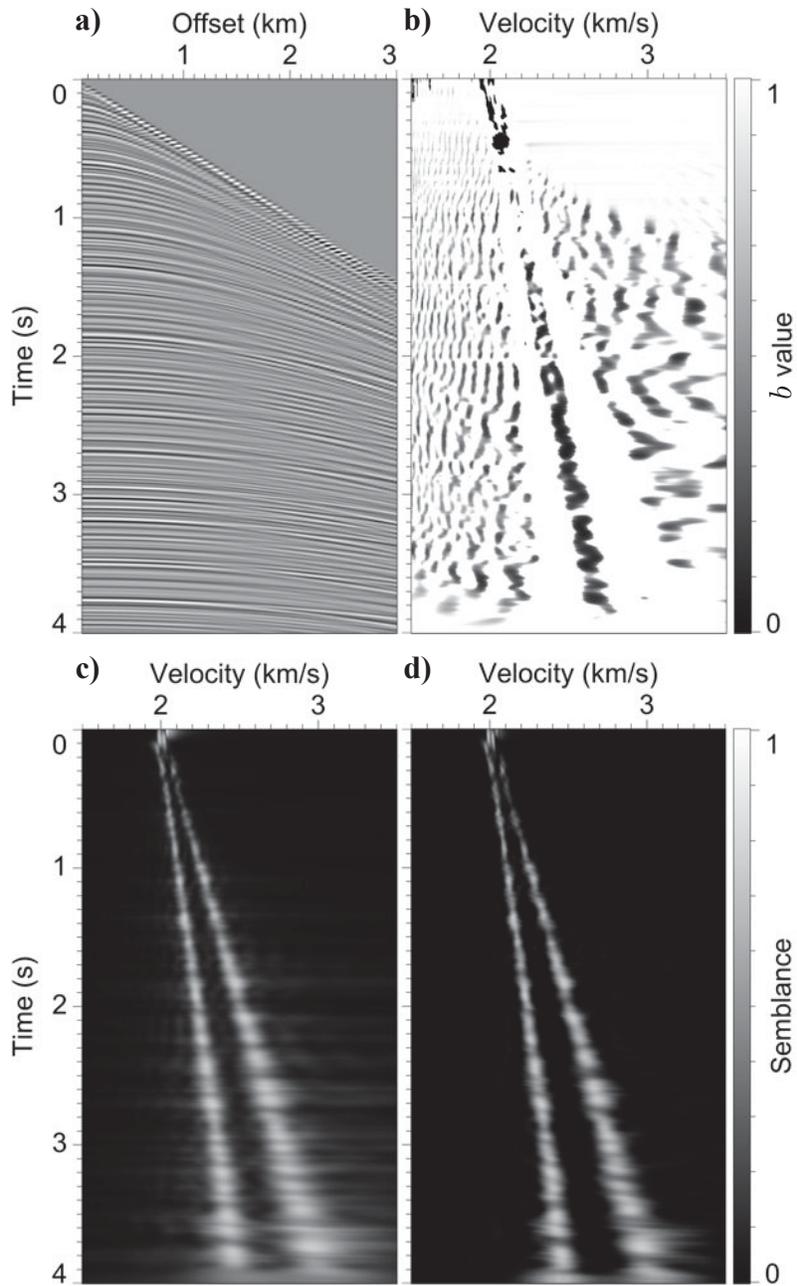


Figure 2.3: Synthetic CMP gather (a), plot of  $b$  values used in the weighting function (b), conventional semblance (c), and weighted semblance (d).

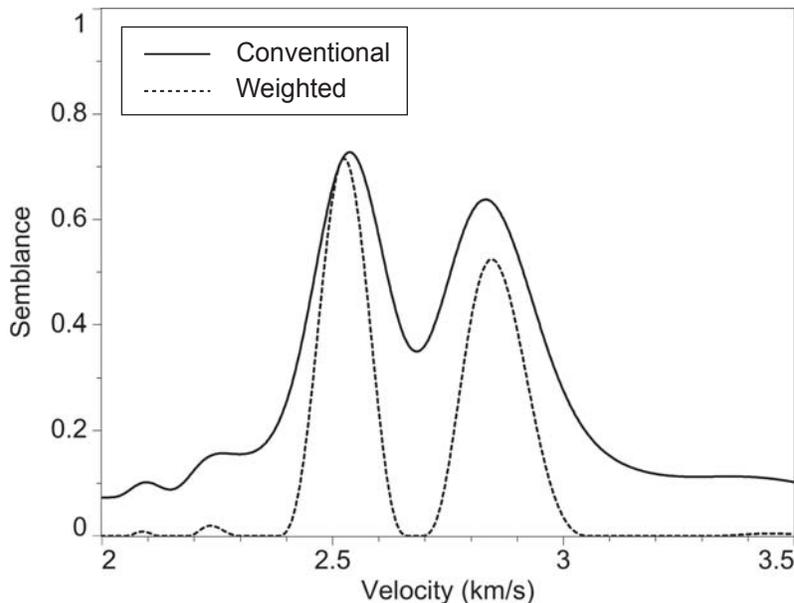


Figure 2.4: Semblance as a function of trial velocity at  $\tau = 3.2$  s, for the semblance spectra shown in Figure 2.3.

For our last synthetic example, we contaminate the synthetic gather shown in Figure 2.3(a) with additive noise. We add bandlimited random noise to the CMP gather with a signal-to-noise ratio of 1, where the signal-to-noise ratio is computed as the ratio of the root-mean-square (rms) amplitude of the signal to the rms amplitude of the noise. Figure 2.5(a) depicts the noise-contaminated synthetic CMP gather, and Figure 2.5(b) depicts the  $b$  values used in the weighting function. Figure 2.5(c) and Figure 2.5(d) show the conventional and weighted semblance spectrum, respectively. Compared to the noise-free synthetic (Figure 2.3), the semblance values overall are lower because of the additive noise, but again, as evidenced by the semblance curve in Figure 2.6, we see an increase in resolution and a decrease in the range of semblance values going from conventional to weighted semblance.

To compare the accuracy of conventional and weighted semblance, we generate 1000 different synthetic CMP gathers with the same NMO velocity and signal-to-noise ratio as the gather shown in either Figure 2.3(a) or Figure 2.5(a), and we compute the error in NMO velocity as the difference between the NMO velocity of the picked semblance peaks and the correct NMO velocity. From these errors, we compute the rms error (for all 1000 gathers)

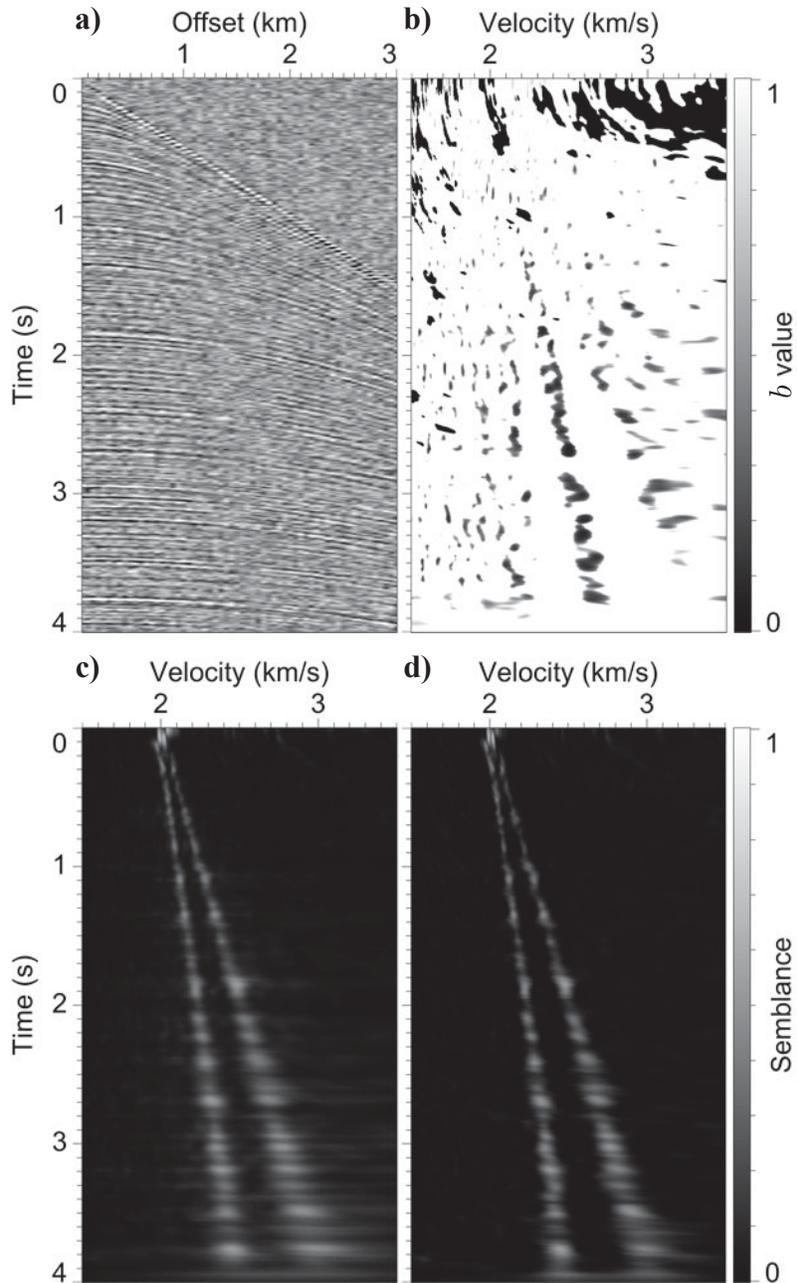


Figure 2.5: Synthetic CMP gather (a), plot of  $b$  values used in the weighting function (b), conventional semblance (c), and weighted semblance (d).

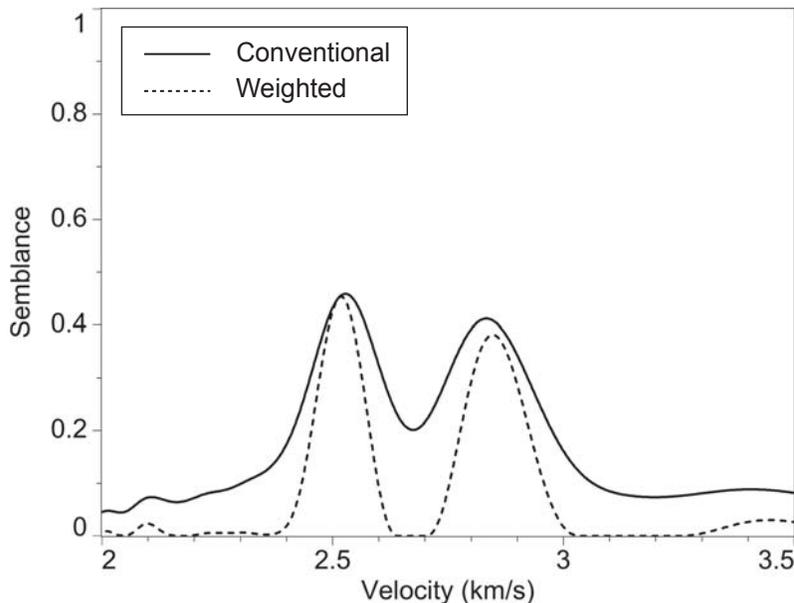


Figure 2.6: Semblance as a function of trial velocity at  $\tau = 3.2$  s, for the semblance spectra shown in Figure 2.5.

as a function of zero-offset time. Figure 2.7(a) shows the rms error of primary reflections for CMP gathers with no noise, while Figure 2.7(b) shows the error of primary reflections for gathers with a signal-to-noise ratio of 1. In the figure, the gray and black lines correspond to conventional and weighted semblance, respectively. For the primary reflections (and also for the multiple reflections, which are not shown), the rms errors for weighted semblance generally are less than that of conventional semblance. However, keep in mind that these rms errors are computed from 1000 different gathers, and so for any single gather, we may expect to see smaller or larger errors for conventional and weighted semblance. That said, these results suggest that at least for gathers similar to our synthetic CMP gathers (i.e., gathers with no amplitude or phase variation with offset and with perfect hyperbolic moveout), weighted semblance is overall more accurate than conventional semblance.

#### 2.4.2 Field data examples

To compare conventional and weighted semblance for real seismic data, we selected four shot gathers from *Seismic Data Analysis* by Oz Yilmaz (2001). These data were selected

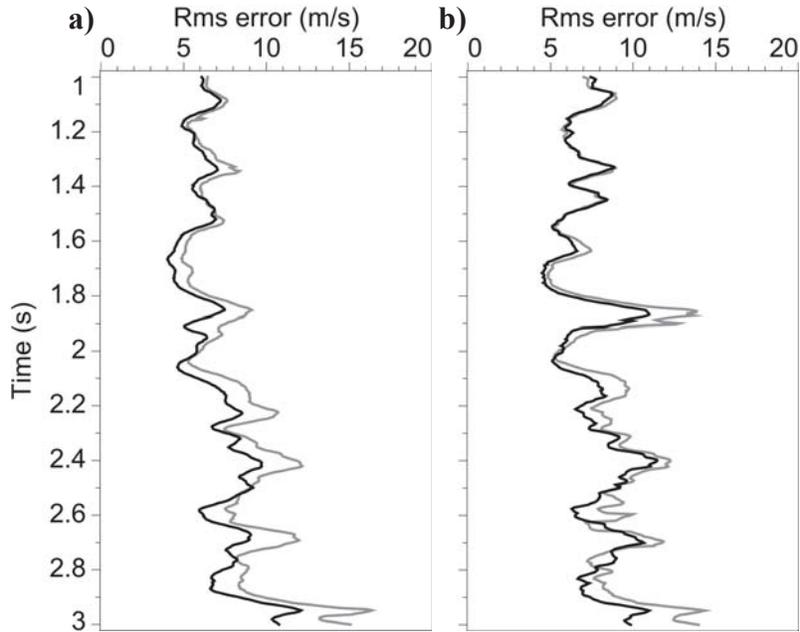


Figure 2.7: Root-mean-square error of NMO velocities of semblance peaks for noise-free synthetic CMP gathers (a), and for synthetic CMP gathers with signal-to-noise ratio equal to one (b). Gray and black lines correspond to conventional and weighted semblance, respectively.

because they are publicly available, and as such they allow our examples to be easily reproduced. Note that because these examples are shot gathers rather than CMP gathers, the NMO correction applied prior to the semblance calculation is not entirely appropriate. However, the comparison between conventional and weighted semblance is still valid.

Figure 2.8(a) shows Yilmaz’s shot record #3, which contains reflections with near-perfect hyperbolic moveout; Figure 2.9(a) shows shot record #8, which contains significant linear noise; Figure 2.10(a) shows shot record #16, which contains long-period multiples between 1 and 3 s; and Figure 2.11(a) shows shot record #30, which is a deep-water shot record containing strong first-order water bottom multiples (Yilmaz, 2001). Figure 2.8(b), Figure 2.9(b), Figure 2.10(b), and Figure 2.11(b) show the  $b$  values used in the weighting function; Figure 2.8(c), Figure 2.9(c), Figure 2.10(c), and Figure 2.11(c) show the conventional semblance spectra; and Figure 2.8(d), Figure 2.9(d), Figure 2.10(d), and Figure 2.11(d) show the weighted semblance spectra. Again, we see that the weighted semblance spectra have higher resolution compared to the conventional semblance spectra, and also that the

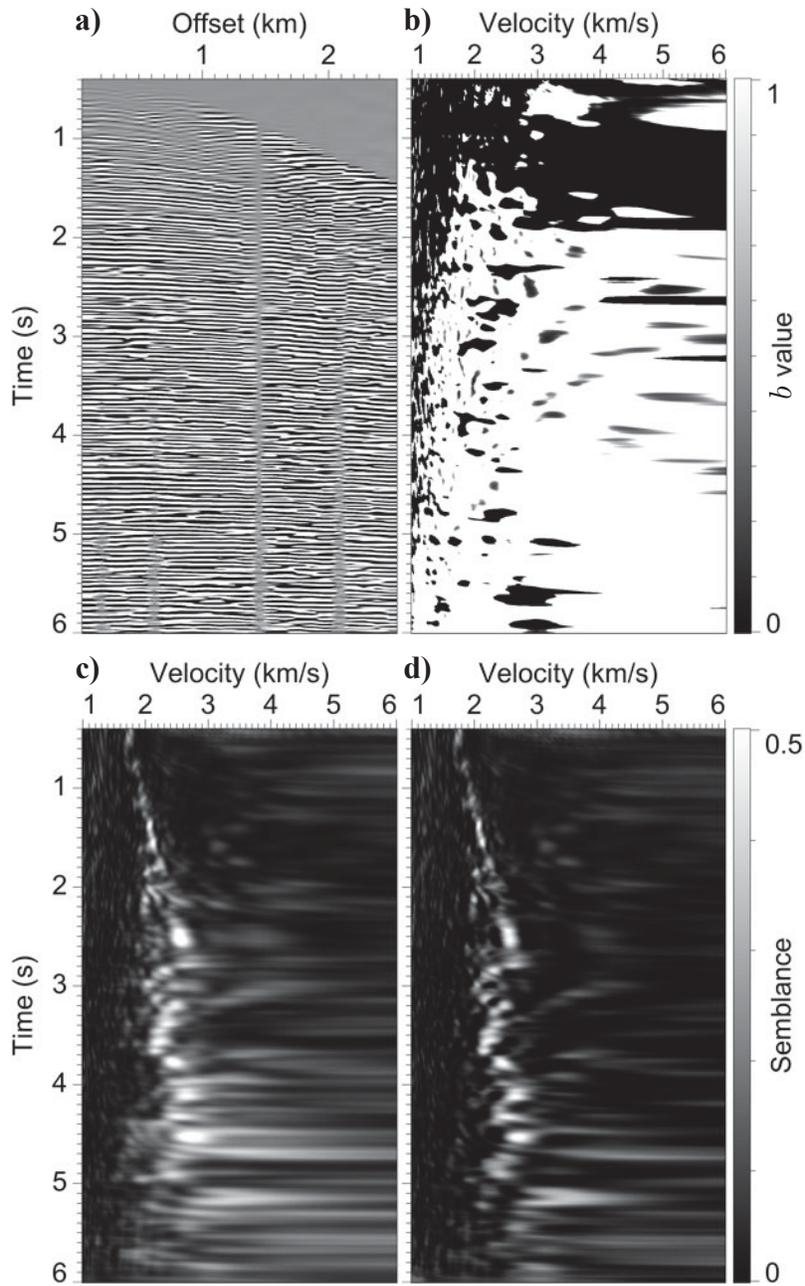


Figure 2.8: Shot record #3 from *Seismic Data Analysis* (a), plot of  $b$  values used in the weighting function (b), conventional semblance (c), and weighted semblance (d).

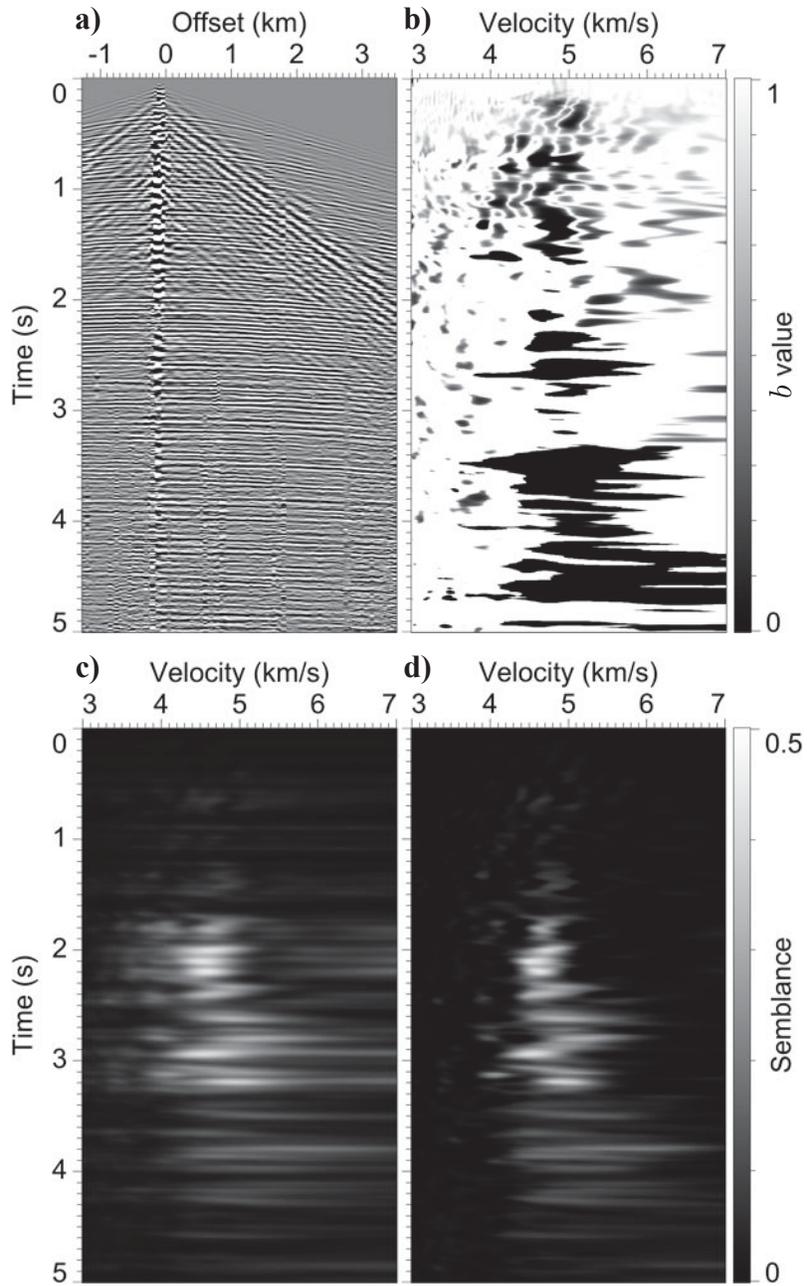


Figure 2.9: Shot record #8 from *Seismic Data Analysis* (a), plot of  $b$  values used in the weighting function (b), conventional semblance (c), and weighted semblance (d).

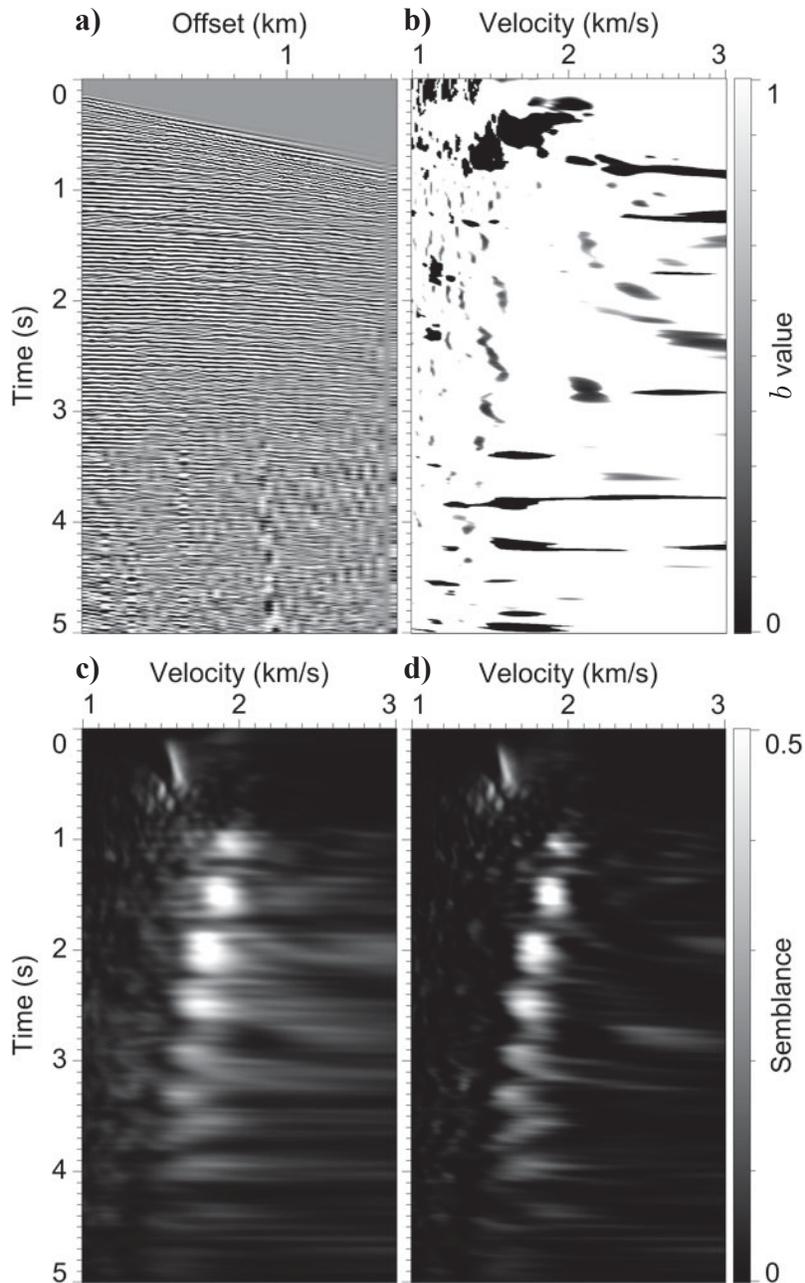


Figure 2.10: Shot record #16 from *Seismic Data Analysis* (a), plot of  $b$  values used in the weighting function (b), conventional semblance (c), and weighted semblance (d).

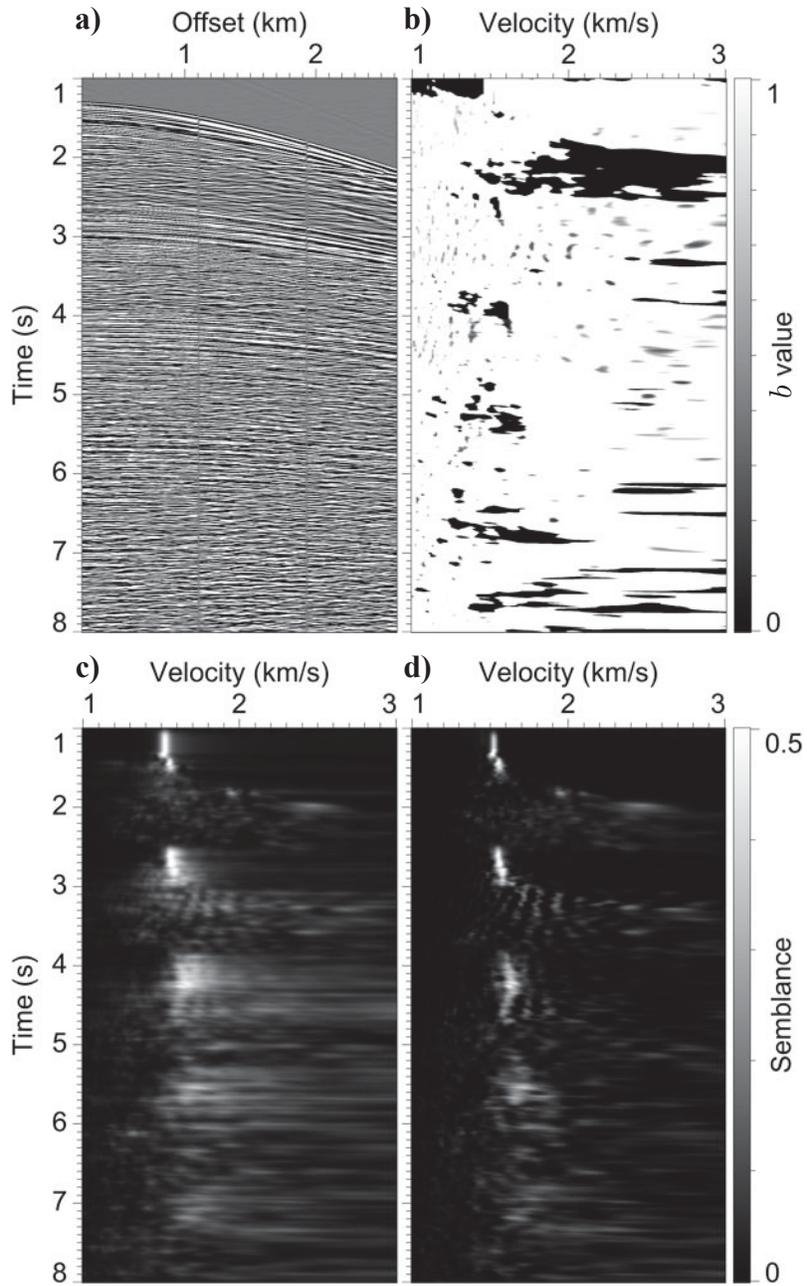


Figure 2.11: Shot record #30 from *Seismic Data Analysis* (a), plot of  $b$  values used in the weighting function (b), conventional semblance (c), and weighted semblance (d).

locations of corresponding semblance peaks remain mostly unchanged. However, for these examples, it is difficult to assess the accuracy of the semblance spectra, since we do not know the correct velocity.

It is worth noting that for both the field data and synthetic data examples, the  $b$  values that minimize semblance are mostly nonzero, and moreover tend to be closer to one, indicating that our algorithm prefers fully-weighted semblance over conventional semblance. This seems to confirm that minimizing semblance indeed increases resolution, since far offsets are more sensitive to changes in NMO velocity and thus can potentially afford higher resolution.

## 2.5 Conclusion

Our implementation of weighted semblance increases the resolution of semblance spectra by using a weighting function to minimize semblance while maintaining a normalized semblance value bounded between zero and one.

Minimizing semblance increases the resolution of semblance spectra because semblance peaks for which the semblance value equals one are not affected by the weighting function. For semblance peaks with semblance value less than one, minimizing semblance does decrease the peak value, but to a lesser extent than it does for semblance peaks closer to zero. Because we are minimizing semblance, the weighted semblance value will never exceed the corresponding conventional semblance value, and for this reason, weighted semblance will not produce new semblance peaks that are not seen in the conventional semblance spectrum—it will only increase the resolution of existing peaks.

Implementing the weighted semblance calculation requires a small change to the conventional semblance implementation. The cost of computing weighted semblance is at least twice that of conventional semblance, but their costs are still comparable, as the computational complexity of calculating weighted semblance remains on the order of  $N_x \times N_t \times N_v$ , where  $N_x$ ,  $N_t$ , and  $N_v$  are the number of offset, time, and velocity samples, respectively. Moreover, if one computes weighted semblance, it is easy to simultaneously compute conventional semblance for little additional cost.

Finally, it is worth noting that the use of our weighted semblance coefficient does not preclude the use of other modified semblance coefficients that perform better in cases of, say, signal amplitude or phase variation with offset. In fact, our method for increasing resolution by minimizing semblance could be extended to work in conjunction with these modified semblance coefficients, by formulating a coefficient that measures correlation with a trend and also increases resolution by minimizing semblance.

## **2.6 Acknowledgements**

Thanks to the sponsors of the Center for Wave Phenomena at the Colorado School of Mines. Thanks also to the reviewers of this paper, including Ken Lerner, Jeff Godwin, Debashish Sarkar, and another anonymous reviewer, for many helpful comments and suggestions.

## CHAPTER 3

### LEAST-SQUARES MIGRATION IN THE PRESENCE OF VELOCITY ERRORS

Modified from a paper published in *Geophysics*, 2014, 79 (4), S153-S161

Simon Luo<sup>1</sup> and Dave Hale<sup>1</sup>

#### 3.1 Summary

Seismic migration requires an accurate background velocity model that correctly predicts the kinematics of wave propagation in the true subsurface. Least-squares migration, which seeks the inverse rather than the adjoint of a forward modeling operator, is especially sensitive to errors in this background model, which can result in traveltimes differences between predicted and observed data that lead to incoherent and defocused migration images. We propose an alternative misfit function for use in least-squares migration that measures amplitude differences between predicted and observed data, i.e., differences after correcting for nonzero traveltimes shifts between predicted and observed data. We demonstrate on synthetic and field data that, when the background velocity model is incorrect, the use of this misfit function results in better focused migration images. Results suggest that our method best enhances image focusing when differences between predicted and observed data can be explained by traveltimes shifts.

#### 3.2 Introduction

Seismic migration can be described as the adjoint of a linearized forward modeling operator applied to observed data (Claerbout, 1992). Migration produces a reflectivity image, an image of a perturbation to the background velocity model (Cohen & Bleistein, 1979), that approximates the true reflectivity insofar as the adjoint of the forward operator approximates the pseudoinverse. Typically, the adjoint is a poor approximation, and the accuracy

---

<sup>1</sup> Center for Wave Phenomena, Department of Geophysics, Colorado School of Mines

of the computed reflectivity image can be significantly improved by using the pseudoinverse of the forward operator rather than the adjoint. The use of the pseudoinverse of the forward operator in migration is known as least-squares migration (Dai, 2012; Köhl & Sacchi, 2003; Nemeth *et al.*, 1999; Østmo & Plessix, 2002; Plessix & Mulder, 2004).

Least-squares migration requires the inverse of the Hessian matrix (the normal operator) of second derivatives of a misfit function with respect to model parameters. The Hessian, however, is prohibitively expensive to compute and store for most practical-sized problems. Approximations of the inverse Hessian (Chavent & Plessix, 1999; Gray, 1997; Guitton, 2004; Plessix & Mulder, 2004; Rickett, 2003; Shin *et al.*, 2001; Symes, 2008; Valenciano, 2008) are more feasible, and are often used to improve the quality of final migration images or to precondition iterative least-squares migration. In this paper, we focus on iterative least-squares migration, which can be used in conjunction with or in place of approximations of the Hessian. An advantage of iterative migration algorithms is that they typically are straightforward to implement; a disadvantage is that they can be more computationally expensive compared to an efficient approximation of the Hessian, or compared to a single application of the adjoint operator as is done, for example, in reverse-time migration (Baysal *et al.*, 1983; Levin, 1984; Loewenthal & Mufti, 1983; McMechan, 1983; Whitmore, 1983).

The quality and accuracy of migration images depends greatly on the accuracy of the background velocity model, and errors in this background model can lead to an incoherent, defocused image. Ideally, the background velocity model should correctly predict the traveltimes of observed data, and should be sufficiently smooth so as not to generate reflected waves. These requirements derive from the conditions under which the Born approximation is valid (Symes, 2009), and under these conditions, migration can accurately image subsurface structures. However, when these conditions are violated, migration images are degraded and become defocused and incoherent. One reason for this degradation is that migration inverts for the perturbation to the background velocity model that controls only the amplitudes of predicted data; if the background model contains errors, then the predicted data

will contain errors in both traveltime and amplitude compared to the observed data, and both these types of errors — instead of only the amplitude errors — will contribute to the migration image.

Often, separating these types of errors, and perhaps discarding a certain type of error, can improve inversion results. For example, for full waveform inversion (Pratt *et al.*, 1998; Tarantola, 1984), authors advocate using only phase or traveltime information (Bednar *et al.*, 2007; Choi & Alkhalifah, 2011; Kamei *et al.*, 2011; Shin & Min, 2006), especially to update the low-wavenumber background model that is difficult for full waveform inversion to recover from reflection seismic data (Hicks & Pratt, 2001; Ma, 2012; Snieder *et al.*, 1989; Xu, 2012). Our task in least-squares migration is complementary to that of full waveform inversion for the background model: we seek to invert for the high-wavenumber component of the model, i.e., the perturbation to the background model. Thus, analogous to the use of phase or traveltime information to recover the low-wavenumber component of the velocity model, we propose to use amplitude information to recover the high-wavenumber component.

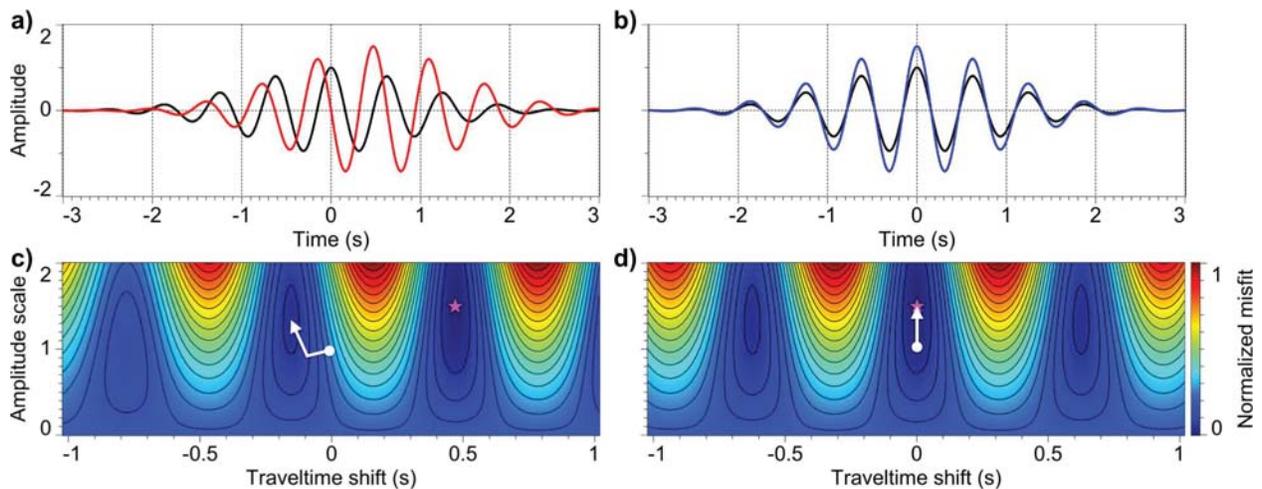


Figure 3.1: A simple example. The (a) predicted data (black) and observed data (red); (b) predicted data (black) and shifted observed data (blue); (c) normalized misfit function computed with the predicted and observed data shown in (a); and (d) normalized misfit function computed with the predicted and shifted observed data shown in (b). A local optimization method beginning at the position indicated by the white circle will converge to a local minimum in (c), but will find the global minimum, indicated by the magenta star, in (d).

The utility of this amplitude and traveltime separation is easily illustrated. Consider the task of estimating a traveltime shift and an amplitude scale between two 1D signals, shown in Figure 3.1. In Figure 3.1a and Figure 3.1b, the black curve represents the predicted data, the red curve represents the observed data, and the blue curve represents the observed data shifted so that its traveltime matches that of the predicted data. The conventional least-squares misfit function, i.e., the  $L^2$ -norm of the difference between predicted and (possibly shifted) observed data, for two model parameters (the amplitude scale and the traveltime shift) is shown in Figure 3.1c and Figure 3.1d. Notice in Figure 3.1c and Figure 3.1d the location of the global minimum, indicated by the magenta star. Figure 3.1c shows the normalized misfit function computed between predicted and observed data (Figure 3.1a). With this misfit function, a local optimization method beginning at the location of the white circle will descend to a local minimum, indicated by the white arrow. In comparison, the normalized misfit function computed between predicted and *shifted* observed data (Figure 3.1b) shown in Figure 3.1d enables the global minimum to be found.

This simple example suggests that by first correcting for traveltime shifts between predicted and observed data, we are better able to estimate the component of the model that controls the amplitude (in this case, the amplitude scale). This is the approach we wish to pursue for the more complicated problem of least-squares migration. The simple example relied on the fact that we could correctly estimate the traveltime shift between predicted and observed data. For migration, we require an accurate and robust method for estimating traveltime shifts between two (not necessarily 1D) signals, and for this purpose we use dynamic warping (Hale, 2013a).

We propose a simple modification of the conventional least-squares misfit function used in iterative least-squares migration. Rather than minimize the difference between predicted and observed data, we propose to minimize their difference after correcting for nonzero traveltime shifts. Assuming estimated traveltime shifts between predicted and observed data are accurate, this misfit function quantifies mostly amplitude differences. We demonstrate

that the use of this amplitude misfit function in least-squares migration results in more coherent and better focused images when the background velocity model used for migration differs from the true background velocity model.

### 3.3 Methods

In this section, we first briefly review linearized waveform inversion and then discuss dynamic warping, the method we use to estimate traveltimes shifts, before presenting our method for amplitude-only inversion.

#### 3.3.1 Linearized waveform inversion

Wave propagation in the subsurface is described approximately by the constant-density acoustic wave equation,

$$\sigma_0 \frac{\partial^2 u_0}{\partial t^2} - \Delta u_0 = f, \quad (3.1)$$

where  $u_0$  is the wavefield,  $\sigma_0$  is the squared background slowness, and  $f$  is the source function. Perturbing  $\sigma_0$  by a scattering potential  $m$  and linearizing about  $m$  yields

$$\sigma_0 \frac{\partial^2 u}{\partial t^2} - \Delta u = -m \frac{\partial^2 u_0}{\partial t^2}, \quad (3.2)$$

where  $u$  is the scattered or perturbation wavefield. Often  $m$  is referred to as the reflectivity model or simply the reflectivity.

Let  $\mathbf{u}_s$  denote the discretized solution of equation 3.2 for a source function at position  $s$ . The wavefield  $\mathbf{u}_s$  is linear in the reflectivity  $\mathbf{m}$ :

$$\mathbf{u}_s = \mathbf{L}_s \mathbf{m}, \quad (3.3)$$

where  $\mathbf{L}_s$  is a linear prediction operator describing the evolution of the scattered wavefield in equation 3.2. The predicted data  $\mathbf{p}_{s,r}$  are a subset of the wavefield  $\mathbf{u}_s$ :

$$\mathbf{p}_{s,r} = \mathbf{S}_r \mathbf{u}_s, \quad (3.4)$$

where  $\mathbf{S}_r$  is a sampling operator that extracts the wavefield at receiver position  $r$ .

To solve equation 3.4 for the reflectivity model  $\mathbf{m}$ , we minimize, in a least-squares sense, the difference between predicted data  $\mathbf{p}_{s,r}$  and observed data  $\mathbf{d}_{s,r}$ :

$$\min_{\mathbf{m}} J(\mathbf{m}) = \sum_{s,r} E_{s,r}(\mathbf{u}_s(\mathbf{m})), \quad (3.5)$$

where

$$E_{s,r}(\mathbf{u}_s) = \frac{1}{2} \|\mathbf{S}_r \mathbf{u}_s - \mathbf{d}_{s,r}\|^2. \quad (3.6)$$

To minimize equation 3.5, we can pursue the negative of the gradient direction

$$\frac{\partial J}{\partial \mathbf{m}} = \sum_{s,r} \mathbf{L}_s^\top \left( \frac{\partial E_{s,r}}{\partial \mathbf{u}_s} \right), \quad (3.7)$$

where

$$\frac{\partial E_{s,r}}{\partial \mathbf{u}_s} = \mathbf{S}_r^\top (\mathbf{S}_r \mathbf{u}_s - \mathbf{d}_{s,r}) \quad (3.8)$$

is the data residual. The adjoint of the prediction operator  $\mathbf{L}_s$  is a migration operator (Claerbout, 1992), and so we obtain the well-known result (Lailly, 1983; Tarantola, 1984) that the gradient of the least-squares misfit function can be computed by a migration of the residuals.

### 3.3.2 Dynamic warping

Before we can consider an amplitude misfit function, we require a method for estimating time-varying traveltimes between predicted and observed data. For this purpose, we use dynamic warping (Hale, 2013a). Dynamic warping is robust and remains accurate in the presence of noise, and compared to more conventional methods for estimating traveltimes based on windowed crosscorrelations, dynamic warping is more accurate, especially when traveltimes vary rapidly as a function of time (Hale, 2013a).

Dynamic time warping (Sakoe & Chiba, 1978) is a method for computing integer time shifts  $\boldsymbol{\tau} = (\tau_1, \tau_2, \dots, \tau_n)$  between two sequences  $\mathbf{p} = (p_1, p_2, \dots, p_n)$  and  $\mathbf{d} = (d_1, d_2, \dots, d_n)$  such that

$$A = \frac{1}{2} \sum_i (p_i - d_{i+\tau_i})^2 \quad (3.9)$$

is minimized with respect to  $\boldsymbol{\tau}$  subject to the constraint

$$|\tau_i - \tau_{i-1}| \leq 1/c, \quad (3.10)$$

where  $c$  is a positive integer. An attractive feature of dynamic time warping is that the algorithm is guaranteed to find the traveltimes shifts  $\boldsymbol{\tau}$  that minimize equation 3.9 subject to constraint 3.10, and these shifts are such that  $\partial A / \partial \boldsymbol{\tau} = \mathbf{0}$  when the constraint is inactive.

Although we could use dynamic time warping to independently estimate traveltimes shifts between all pairs of predicted and observed traces, in practice we find that using dynamic image warping (Hale, 2013a) to estimate traveltimes shifts between predicted and observed common shot gathers yields more accurate shifts, especially when predicted and observed data are not simply shifted versions of each other (as is often the case even with synthetic data, and certainly always is the case with recorded field data). Dynamic image warping (Hale, 2013a) approximately solves the extension to higher dimensions of the constrained optimization problem specified by equations 3.9 and 3.10, and in doing so, imposes constraints both in time (equation 3.10) as well as in distance or offset on the estimated traveltimes shifts.

### 3.3.3 Inversion of amplitude errors

To formulate an inversion of amplitude errors, we modify the observed data to include a time-shift operator:

$$\mathbf{b}_{s,r} = \mathbf{T}_{s,r} \mathbf{d}_{s,r}, \quad (3.11)$$

where  $\mathbf{T}_{s,r}$  is a linear operator, e.g., a sinc interpolation operator, that shifts the observed data  $\mathbf{d}_{s,r}$  by the traveltimes shifts  $\boldsymbol{\tau}_{s,r}$  estimated using dynamic warping. Note that  $\mathbf{T}_{s,r}$  depends implicitly on the model  $\mathbf{m}$ , because the traveltimes shifts  $\boldsymbol{\tau}_{s,r}$  are computed using

the predicted data  $\mathbf{p}_{s,r}$ , which depend on the model.

The shifted observed data  $\mathbf{b}_{s,r}$  can be viewed as a secondary dataset obtained by processing the observed data. Processing of the observed data prior to migration is standard practice, even for conventional migration. The purpose of this processing is essentially to remove from the observed data any components that are due to an inconsistent model of wave propagation in the true subsurface. For example, just as an acoustic wave equation cannot explain shear waves in elastic data, the linearized wave equation (equation 3.2) with an incorrect background model cannot explain the traveltimes of the observed data. Migration using an incorrect background model is equivalent to migration using forward modeling that is inconsistent with the observed data, and so to properly migrate these data, we must first remove those components that cannot be explained by our forward modeling. Those components are the traveltimes.

Thus, we seek to minimize the difference between predicted data  $\mathbf{p}_{s,r}$  and time-shifted observed data  $\mathbf{b}_{s,r}$ :

$$\min_{\mathbf{m}} J_A(\mathbf{m}) = \sum_{s,r} A_{s,r}(\mathbf{u}_s(\mathbf{m}), \boldsymbol{\tau}_{s,r}(\mathbf{m})), \quad (3.12)$$

where

$$A_{s,r} = \frac{1}{2} \|\mathbf{S}_r \mathbf{u}_s - \mathbf{T}_{s,r} \mathbf{d}_{s,r}\|^2. \quad (3.13)$$

Note that if the estimated traveltime shifts  $\boldsymbol{\tau}_{s,r}$  are accurate, then equation 3.12 measures only amplitude errors between predicted and observed data. If the traveltime shifts are zero, then equation 3.12 reduces to equation 3.5.

To minimize the misfit function in equation 3.12, we require its gradient with respect to model parameters:

$$\frac{\partial J_A}{\partial \mathbf{m}} = \sum_{s,r} \mathbf{L}_s^\top \left( \frac{\partial A_{s,r}}{\partial \mathbf{u}_s} \right), \quad (3.14)$$

where

$$\frac{\partial A_{s,r}}{\partial \mathbf{u}_s} = \mathbf{S}_r^\top (\mathbf{S}_r \mathbf{u}_s - \mathbf{T}_{s,r} \mathbf{d}_{s,r}) . \quad (3.15)$$

Although  $A_{s,r}$  depends on the estimated traveltime shifts  $\boldsymbol{\tau}_{s,r}$ , we need not consider this dependence when computing the residual in equation 3.15 because dynamic warping minimizes equation 3.13 (or equation 3.9) subject to constraint 3.10, so that  $\partial A_{s,r} / \partial \boldsymbol{\tau}_{s,r}$  is mostly zero. We refer to equation 3.15 as the amplitude residual and equation 3.12 as the amplitude misfit function, as they measure only amplitude errors between predicted and observed data.

### 3.4 Results

We compare conventional least-squares migration (LSM) with the proposed method of least-squares migration of amplitude errors (LSMA) on a 2D synthetic dataset, and on a 2D field dataset.

For least-squares migration, the data are linear in the reflectivity, and thus LSM images can be computed by minimizing equation 3.5 with (linear) conjugate gradient iterations. To compute LSMA images by solving equation 3.12, however, is a nonlinear problem because the reflectivity  $\mathbf{m}$  depends on the traveltime shifts  $\boldsymbol{\tau}_{s,r}$ , but the traveltime shifts also depend on the reflectivity. We can compute LSMA images either by minimizing equation 3.12 using a gradient-based descent method (e.g., steepest descent or nonlinear conjugate gradient), or alternatively, by first solving equation 3.12 with fixed traveltime shifts  $\boldsymbol{\tau}_{s,r}$ , then recomputing the traveltime shifts and solving equation 3.12 with the new shifts, repeating until convergence.

Note that when the traveltime shifts  $\boldsymbol{\tau}_{s,r}$  are zero, equation 3.12 is equivalent to equation 3.5. This is the case for the first nonlinear iteration or the first solution of equation 3.12 with fixed  $\boldsymbol{\tau}_{s,r}$ , in which the reflectivity is zero and hence the traveltime shifts are zero. After the first nonlinear iteration or the first solution of equation 3.12, we obtain a nonzero reflectivity image from which to predict data and to estimate possibly nonzero traveltime

shifts.

### 3.4.1 Synthetic data example

The background slowness used for modeling and migration is shown in Figure 3.2(b), and is computed by smoothing a modified Marmousi model (Lailly & Versteeg, 1990) shown in Figure 3.2(a) along both the depth and distance axes using a two-sided exponential filter with width 100 m. The true reflectivity shown in Figure 3.2(c) is then computed as the difference between the true slowness (Figure 3.2(a)) squared and the true background slowness (Figure 3.2(b)) squared. Using the true background slowness and true reflectivity, we simulate observed data by solving equations 3.1 and 3.2 for a Ricker source function with peak frequency 10 Hz. To facilitate comparison of LSM and LSMA, all migration images for these synthetic data are computed using 20 nonlinear conjugate gradient iterations. Hence, as the cost of dynamic warping is small compared to the cost of modeling and migration, the LSM and LSMA images computed for these synthetic data come at comparable costs.

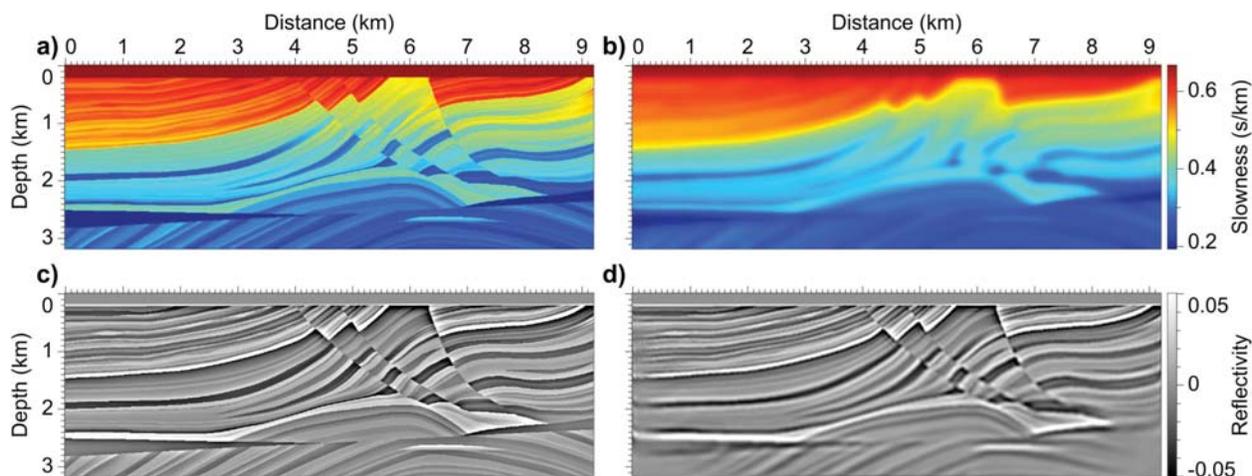


Figure 3.2: The (a) true slowness model, (b) true background slowness model, (c) true reflectivity computed as the difference between the true slowness squared and the true background slowness squared, and (d) LSM image.

The first example shown in Figure 3.2 demonstrates conventional LSM using the true background slowness for migration. The reflectivity image shown in Figure 3.2(d) is obtained

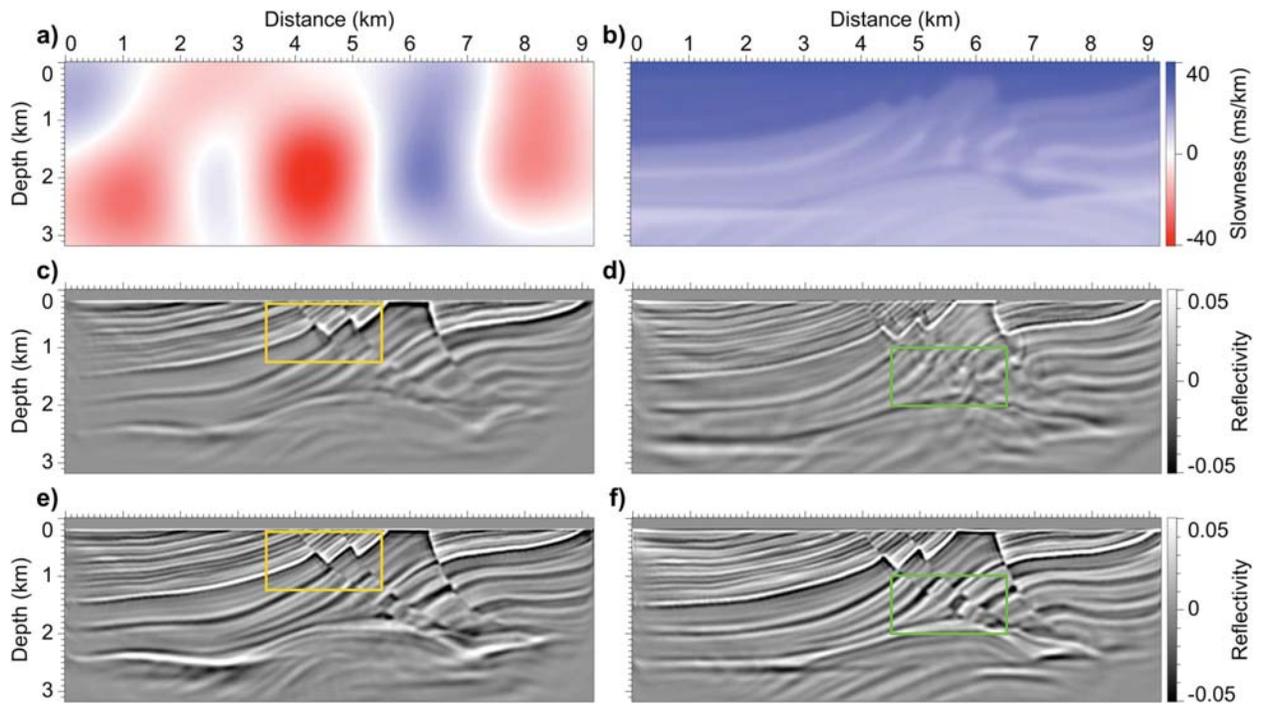


Figure 3.3: The (a,b) difference between the true background slowness shown in Figure 3.2(b) and the background slowness used for migration; (c) LSM image computed for the background slowness with error shown in (a); (d) LSM image computed for the background slowness with error shown in (b); (e) LSMA image computed for the background slowness with error shown in (a); and (f) LSMA image computed for the background slowness with error shown in (b).

after 20 nonlinear conjugate gradient iterations ( Nocedal & Wright, 2000) of LSM using the true background slowness with 153 shots and 767 receivers evenly spaced along the surface. As expected, this computed reflectivity matches well the true reflectivity shown in Figure 3.2(c) because the background slowness model used for migration was exactly the true background slowness. In practice, we expect the background slowness model used to migrate the data to differ from the true background slowness model.

Figure 3.3 illustrates the effects of erroneous background slowness models on the reflectivity images obtained using LSM and LSMA. Figure 3.3(a) and Figure 3.3(b) show the differences between the true background slowness model (Figure 3.2(b)) and the background slowness models that we use for migration. The slowness error shown in Figure 3.3(a) was computed by smoothing a random slowness model, while the error shown in Figure 3.3(b) resulted from scaling the true background slowness by 95%.

Figure 3.3(c) and Figure 3.3(d) show the reflectivity images computed using 20 iterations of LSM with the erroneous background slowness models with errors shown in Figure 3.3(a) and Figure 3.3(b), respectively. Compared to the reflectivity image (Figure 3.2(d)) computed using the true background slowness, the image in Figure 3.3(c) is degraded, and shows uneven illumination and defocused reflectors, especially at greater depths where traveltimes errors resulting from the erroneous background slowness are more severe. This degradation is also seen in the reflectivity image shown in Figure 3.3(d). The quality of this image is worse than the image shown in Figure 3.3(c) because the slowness errors (Figure 3.3(b)) all have the same sign, and so traveltimes errors in the predicted data accumulate more quickly than traveltimes errors for data predicted with the slowness model with error shown in Figure 3.3(a).

Figure 3.3(e) and Figure 3.3(f) show the reflectivity images computed with 20 iterations of LSMA. Compared to the conventional LSM images (Figure 3.3(c) and Figure 3.3(d)), the LSMA images show improved illumination of deeper portions of the model, and better focused and more continuous reflectors throughout. For example, Figure 3.4(a), Figure 3.4(b), and Figure 3.4(c) show zoomed views of the areas enclosed by yellow boxes in Figure 3.3(c),

Figure 3.3(e), and Figure 3.2(c), respectively. Compared to the LSM image shown in Figure 3.4(a), reflectors in the LSMA image shown in Figure 3.4(b) are more focused and better match the true reflectivity shown in Figure 3.4(c). Similarly, zoomed views shown in Figure 3.5(a), Figure 3.5(b), and Figure 3.5(c) of the areas enclosed by green boxes in Figure 3.3(d), Figure 3.3(f), and Figure 3.2(c), respectively, demonstrate that even for a large and biased slowness error (Figure 3.3(b)), minimizing the amplitude misfit function yields an interpretable reflectivity image with features that match those apparent in the true reflectivity.

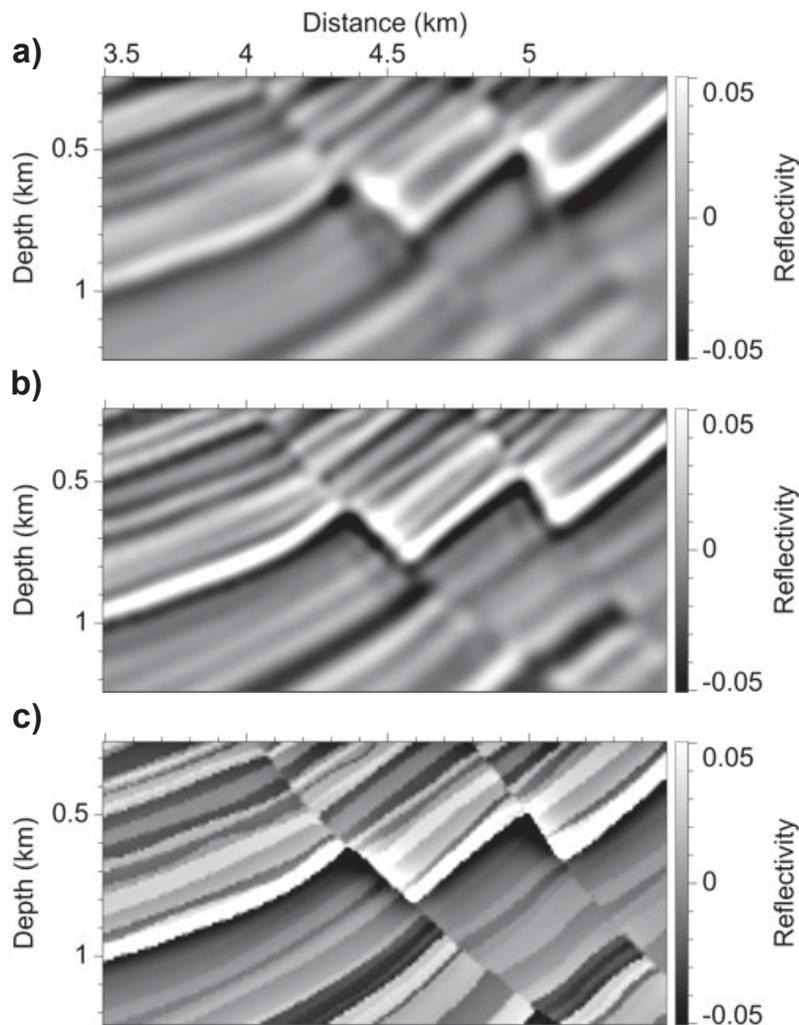


Figure 3.4: Zoomed views of the areas enclosed by yellow boxes in the (a) LSM image in Figure 3.3c, (b) LSMA image in Figure 3.3e, and (c) true reflectivity in Figure 3.2c.

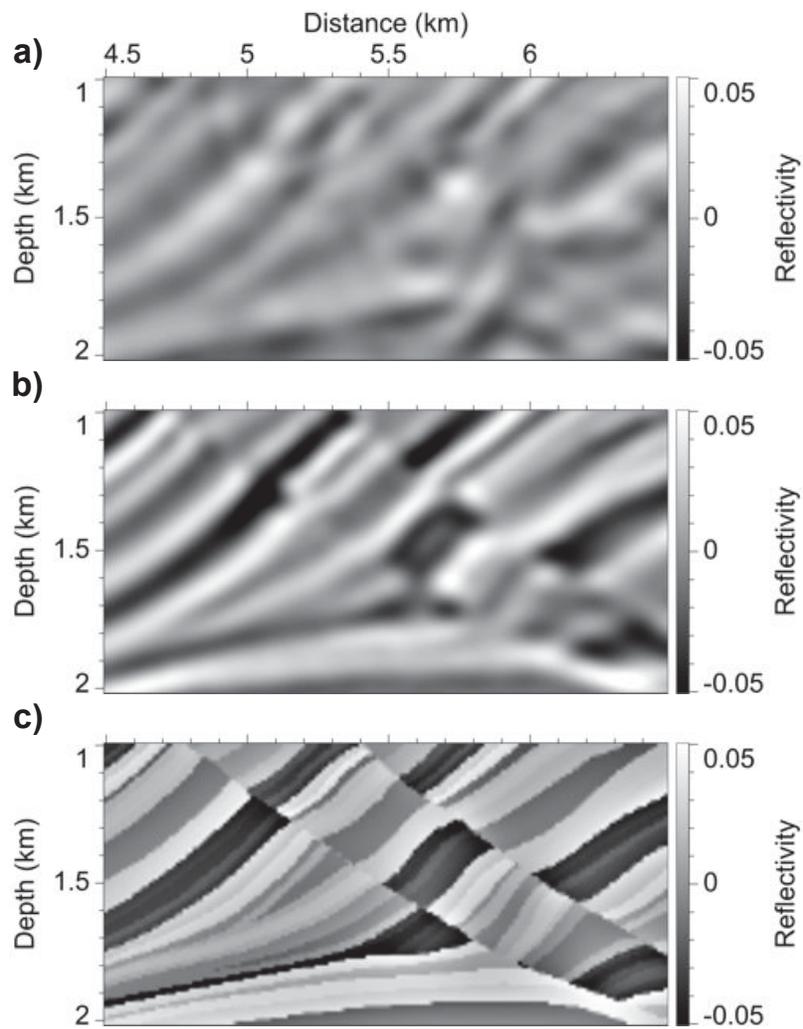


Figure 3.5: Zoomed views of the areas enclosed by green boxes in the (a) LSM image in Figure 3.3d, (b) LSMA image in Figure 3.3f, and (c) true reflectivity in Figure 3.2c.

Note, however, that the positions of features in LSMA images (Figure 3.3(e) and Figure 3.3(f)) are shifted compared to their positions in the true reflectivity (Figure 3.2(c)). For example, compare the position of the reflector located at distance 3.5 km and depth 1 km in Figure 3.4(b), or the reflector located at distance 4.5 km and depth 1 km in Figure 3.5(b), to their positions in the true reflectivity. This mispositioning is expected, however, since LSMA images are computed using erroneous background slowness models.

The presence of remaining traveltimes shifts between predicted and observed data, as well as spatial shifts between image features in computed LSMA images and those in the true reflectivity, is confirmed by the misfit functions shown in Figure 3.6 and Figure 3.7. Figure 3.6(a) and Figure 3.7(a) show normalized data and amplitude misfit functions, while Figure 3.6(b) and Figure 3.7(b) show normalized model misfit functions (the  $L^2$ -norm of the difference between the computed reflectivity and the true reflectivity) for LSM and LSMA images computed using either the true background slowness model shown in Figure 3.2(b) or the erroneous background slowness model with error shown in Figure 3.3(a) or Figure 3.3(b). In Figure 3.6(a) and Figure 3.7(a), note that the data misfit is not used in LSMA, but more importantly, notice that the data misfit increases in iteration 7 in Figure 3.6(a) and in iteration 2 in Figure 3.7(a). This indicates that the better-focused LSMA images shown in Figure 3.3(e) and Figure 3.3(f) *cannot be obtained with conventional LSM*, which minimizes the data misfit.

The model misfits shown in Figure 3.6(b) and Figure 3.7(b) indicate that, for the erroneous background slowness models shown in Figure 3.3(a) and Figure 3.3(b), the LSM images (Figure 3.3(c) and Figure 3.3(d)) more closely match the true reflectivity (Figure 3.2(c)) than do the LSMA images (Figure 3.3(e) and Figure 3.3(f)). Indeed, a zero-reflectivity image is closer to the true reflectivity than the LSMA image shown in Figure 3.3(f). However, the large model misfits for LSMA images simply reflect the fact that features in these images are shifted relative to the corresponding features in the true reflectivity. Although image features in LSMA images are shifted, it is clear that the amplitudes (but not the positions)

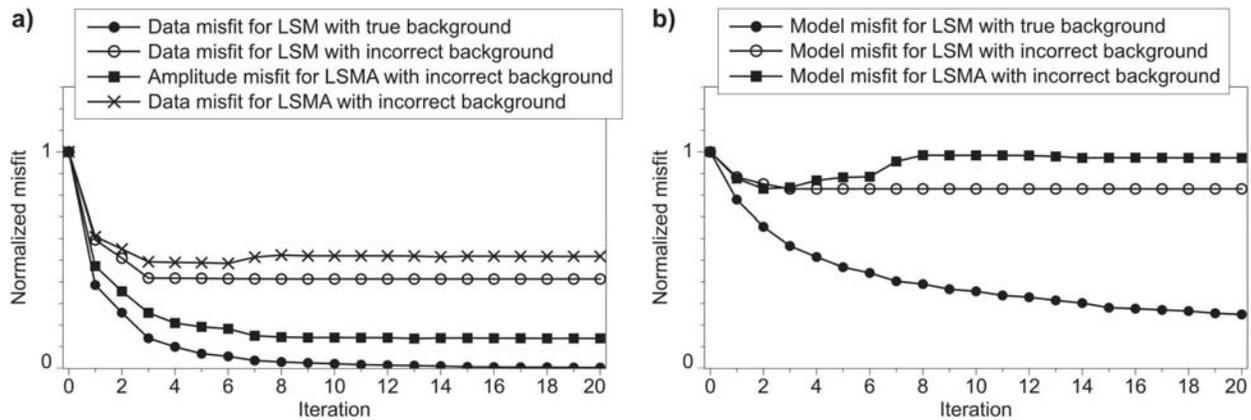


Figure 3.6: Normalized (a) data and amplitude misfit and (b) model misfit for LSM and LSMA. Here, *true background* refers to the true background slowness shown in Figure 3.2(b), while *incorrect background* refers to the background slowness with error shown in Figure 3.3(a).

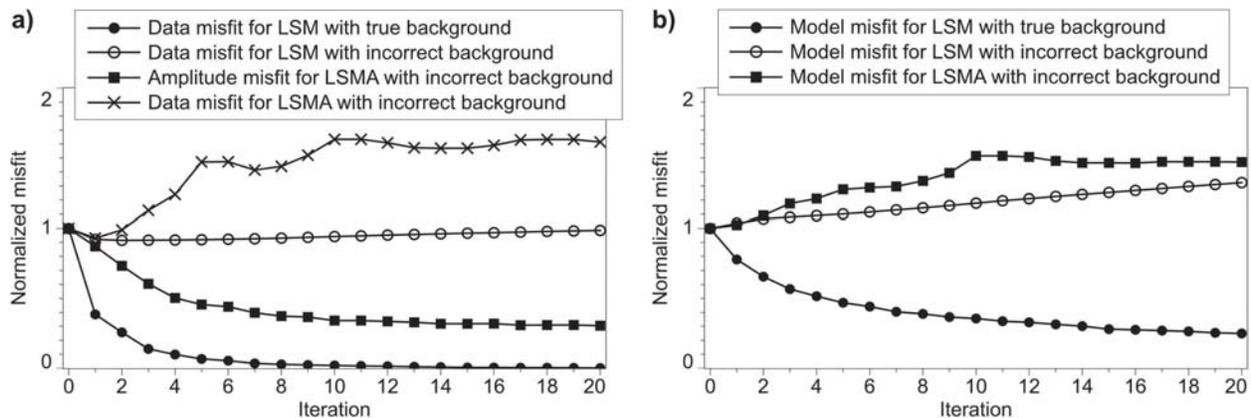


Figure 3.7: Normalized (a) data and amplitude misfit and (b) model misfit for LSM and LSMA. Here, *true background* refers to the true background slowness shown in Figure 3.2(b), while *incorrect background* refers to the background slowness with error shown in Figure 3.3(b).

of these features better match those of the true reflectivity.

### 3.4.2 Field data example

Next we test our method for amplitude-only migration on a subset of a field dataset provided by Eni E&P. The entire 2D dataset contains 3661 shots with a shot spacing of 12.5 m, and was recorded using a streamer with 99 receivers with a receiver spacing of 12.5 m and maximum offset of 1.225 km. The subset of the data that we migrate consists of 431 shots with shot spacing of 25 m. The data have been regularized, and multiples have been attenuated. We estimate a zero-phase wavelet from the amplitude spectrum computed from a subset of the recorded data (Claerbout, 1992), and we apply a bandpass filter to both the estimated wavelet and the recorded data to remove frequency content below 20 Hz and above 80 Hz prior to migration.

We compare LSM and LSMA for two slowness models. The first slowness model, shown in Figure 3.8(a), is laterally invariant (except near the sea floor), while the second, shown in Figure 3.8(b), is an optimized slowness model that was provided with the recorded data. The LSM and LSMA images computed for the laterally invariant slowness model (Figure 3.8(a)) are shown in Figure 3.9(a) and Figure 3.9(b), respectively. Comparing these images, we observe that reflectors in the LSMA image are more continuous and better focused than corresponding reflectors in the LSM image. Moreover, image features in the LSMA image (Figure 3.9(b)) are similar to features seen in the LSM image (Figure 3.9(c)) computed for the optimized slowness model (Figure 3.8(b)), despite the use of a much simpler slowness model for LSMA. Differences between the migration images shown in Figure 3.9(a) and Figure 3.9(b) are most apparent in the areas enclosed by yellow boxes, in which the slowness differences (Figure 3.8(c)) between the models used for migration are relatively large. Zoomed views of the areas enclosed by yellow boxes in Figure 3.9(a), Figure 3.9(b), and Figure 3.9(c) are shown in Figure 3.10(a), Figure 3.10(b), and Figure 3.10(c), respectively. Elsewhere, where slowness errors are smaller, differences between the migration images are less significant, as one would expect.

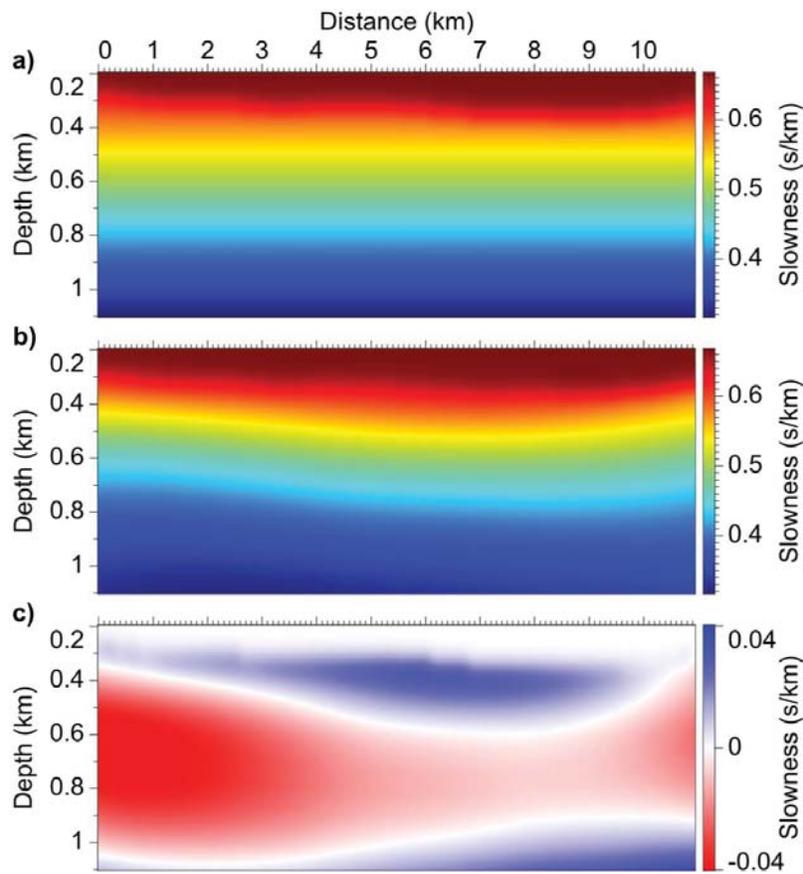


Figure 3.8: The (a) laterally invariant and (b) optimized slowness models used for migration, and the (c) difference between (b) and (a).

It is worth noting that, for this example, it was necessary to use 3D dynamic warping in LSMA. For 3D warping, rather than independently warp predicted to observed shot gathers as was done for the synthetic examples shown in Figure 3.3, we instead warped simultaneously all predicted shot gathers to all observed shot gathers, at each iteration of LSMA. A 3D warping enables us to constrain changes in estimated traveltimes with shot location, which results in more accurate shifts. For synthetic tests in which the same forward modeling code is used to simulate both predicted and observed data, this additional constraint is perhaps unnecessary. For field data, however, an additional constraint on the traveltimes shifts can significantly improve the accuracy of estimated shifts, especially in cases where the data quality is low.

Because we compute LSMA images by minimizing the difference between predicted and shifted observed data (equation 3.12), the predicted data in general will not have the same traveltimes as the original observed data. An example of these traveltimes differences for data corresponding to the shot located at distance 1.85 km is shown in Figure 3.11. Figure 3.11(a) shows the observed data, Figure 3.11(b) shows the predicted data computed using the laterally invariant slowness model (Figure 3.8(a)) and the LSMA image (Figure 3.9(b)), and Figure 3.11(c) shows the traveltimes shifts between the data shown in Figure 3.11(a) and Figure 3.11(b). The maximum frequency content of the data is 40 Hz, which corresponds to a period of 25 ms. Thus we observe from Figure 3.11(c) that the remaining traveltimes shifts between predicted and observed data exceed one half period. This confirms that LSMA yields an image that explains the dynamics, but not the kinematics, of the observed data.

### 3.5 Discussion

The improvement in LSMA images compared to conventional LSM images depends on the nature of the background slowness error, and also on the acquisition geometry. A comparison between the images shown in Figure 3.3(e) and Figure 3.3(f) suggests that LSMA provides a greater improvement in image quality and reflector focusing for small, systematic errors in background slowness (e.g., Figure 3.3(b)), perhaps because in such situations, traveltimes

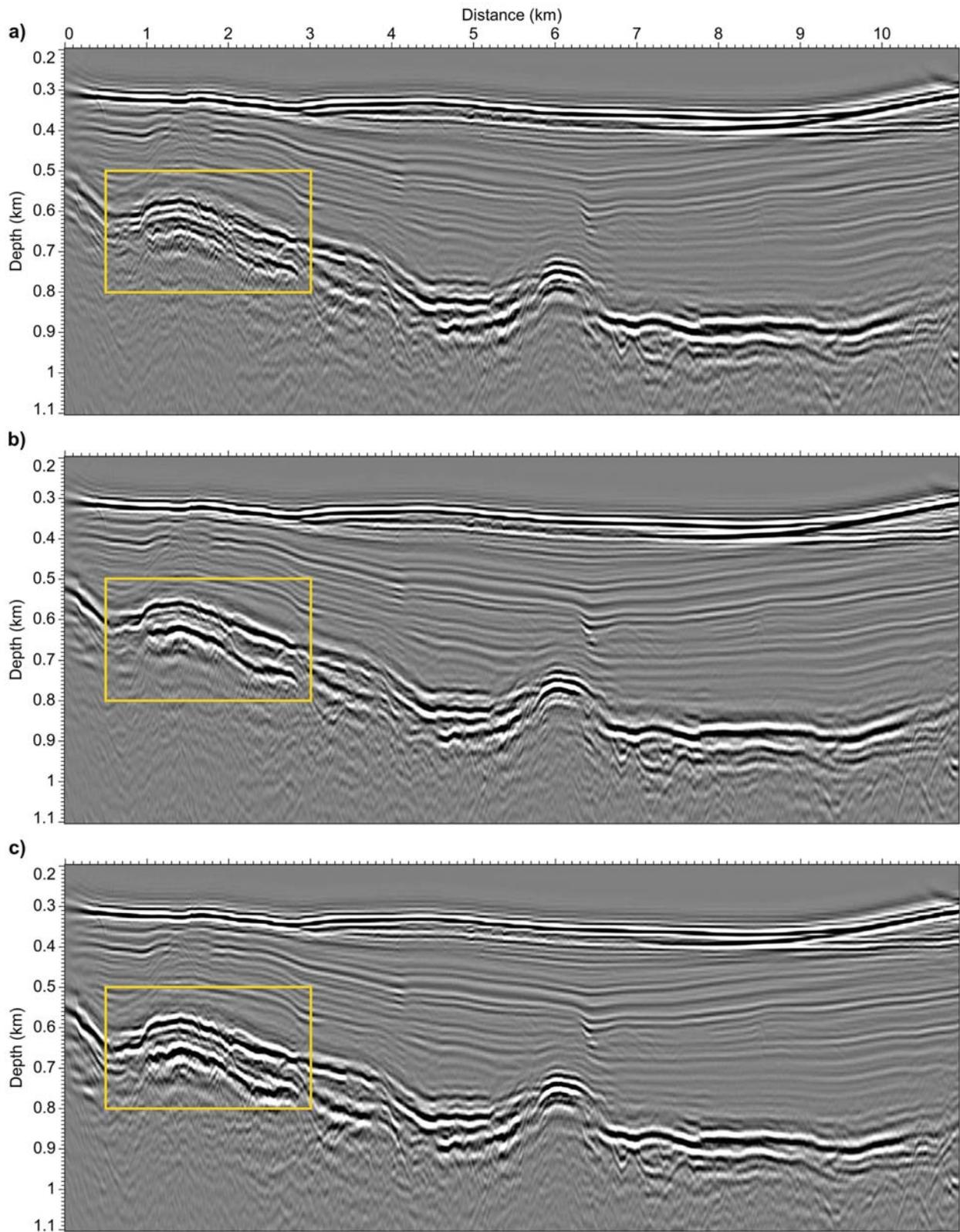


Figure 3.9: The (a) LSM image and (b) LSMA image computed for the laterally invariant slowness model shown in Figure 3.8(a), and the (c) LSM image computed for the optimized slowness model shown in Figure 3.8(b).

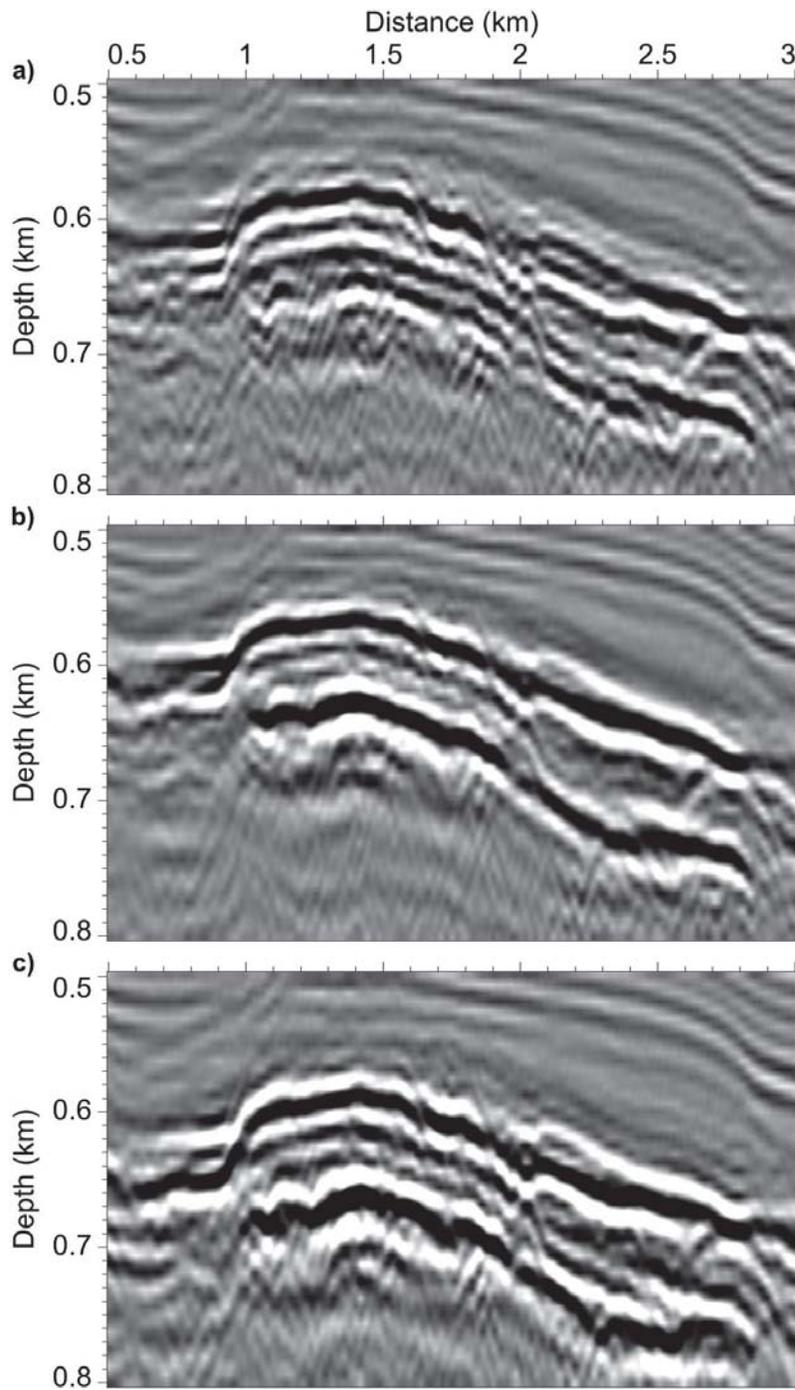


Figure 3.10: Zoomed views of the areas enclosed by yellow boxes in (a) Figure 3.9(a), (b) Figure 3.9(b), and (c) Figure 3.9(c).

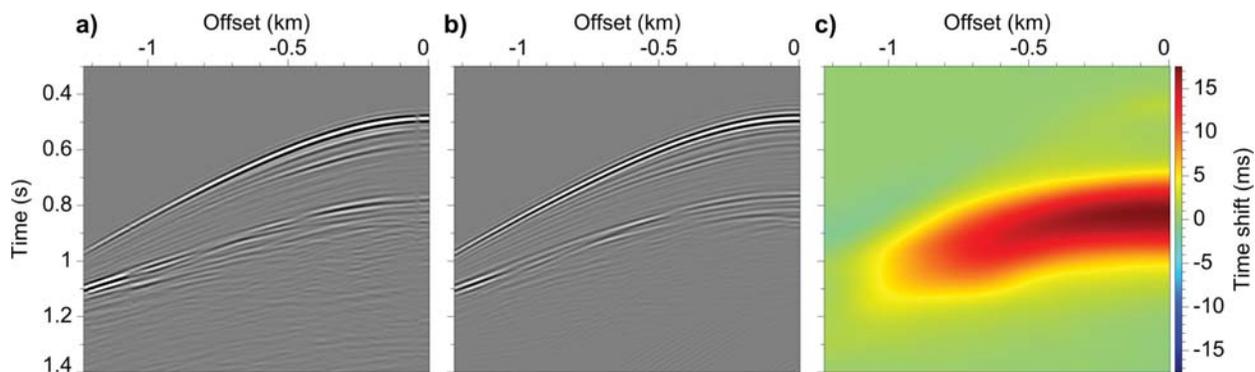


Figure 3.11: For the shot located at distance 1.85 km, the (a) observed data, (b) predicted data computed using the laterally invariant slowness model shown in Figure 3.8(a) and the LSMA image shown in Figure 3.9(b), and (c) traveltime shifts between (a) and (b).

shifts can explain well the differences between predicted and observed data. When the background slowness error is more complex or is too large, predicted and observed data might be inconsistent, i.e., events in one dataset do not have corresponding events in the other, making it difficult to estimate accurate shifts; or, predicted and observed data might differ by significant horizontal spatial shifts in addition to vertical traveltimes shifts, in which case estimating only traveltimes shifts for use in LSMA might be inadequate. While dynamic warping can also be used to estimate horizontal shifts, only vertical traveltimes shifts were used in the examples shown above.

The improvement in LSMA images also depends on acquisition geometry. Traveltimes differences between observed and predicted data in LSM arise from errors in the background slowness model used for migration, but more specifically, they arise from inconsistencies between different images of the same subsurface geologic structures, e.g., images computed for neighboring individual shots. Thus, we expect LSMA to provide greater improvement over conventional LSM when the recorded data provide redundant information about subsurface geologic structures. Conversely, we expect LSMA and LSM images to be more similar when data provide independent information, for example, when shots are sparsely located or shot spacing is large, or when the maximum source-receiver offset or the offset-to-depth ratio is small.

Although LSMA images can provide an improved estimate of the amplitudes of the true reflectivity, ultimately we seek a complete model of the subsurface, which includes not only an accurate reflectivity model but also an accurate background slowness model. The proposed method could potentially be extended and used to aid an inversion for the background slowness. A simple approach might be to hold the reflectivity model constant following LSMA, and then invert the remaining traveltimes shifts between predicted and observed data in order to update the background slowness.

Finally, it is perhaps worth mentioning that a method similar to LSMA can be formulated by introducing a time-shift operator  $\mathbf{T}_{s,r}^{-1}$  to the right-hand-side of equation 3.4, to be applied after computing the predicted data. The resulting forward modeling equation, in which the time-shift operator can be considered a residual modeling operator, leads to an inversion scheme similar to LSMA, the main difference being that the predicted data rather than the observed data are shifted in the computation of the gradient. The resulting image, however, in general differs from the LSMA image (in fact, they are equivalent only in the case of constant shifts). We tested this alternate method on the examples shown above, but the results we obtained were inferior, i.e., image features were less similar to features in the image computed using the best available slowness model, to those obtained with LSMA.

### 3.6 Conclusion

We have presented a method for least-squares migration that minimizes an amplitude misfit function defined with differences between predicted data and shifted observed data, with traveltimes shifts between predicted and observed data estimated using dynamic warping. The use of this amplitude misfit function results in a more coherent and better focused migration image when the background slowness model used for migration contains errors. These LSMA images contain image features with amplitudes that match those of the true reflectivity, but with positions that are shifted relative to the positions of corresponding features in the true reflectivity. LSMA images thus perhaps are better suited for interpretation of geologic structures, but in order to correctly position interpreted structures, we would

need to first correctly position LSMA image features. One way to correct for the mispositioning of image features is to first align features with measurements of subsurface properties obtained from well logs, and then interpolate alignment shifts between well-log locations to generate shifts for an entire image.

### **3.7 Acknowledgement**

This work was supported by the sponsors of the Consortium Project on Seismic Inverse Methods for Complex Structures at the Colorado School of Mines. Thanks to Clément Fleury for helpful discussions and for sharing his insights on this problem. Thanks also to Ivan Vasconcelos and three anonymous reviewers whose comments and suggestions helped to improve the manuscript. The field data were provided courtesy of Eni E&P.

## CHAPTER 4

### NON-VERTICAL DEFORMATIONS FOR SEISMIC IMAGE FLATTENING

Simon Luo<sup>1</sup> and Dave Hale<sup>1</sup>

#### 4.1 Summary

Seismic image flattening produces subsurface images in which sedimentary layering is horizontal. With flattened images, interpretation of stratigraphic features is straightforward, and horizon picking is trivial. Most flattening methods are limited to vertical shearing and stretching of an image. Because of this limitation, these methods may have difficulty flattening seismic images that contain non-vertical deformations without significantly distorting image features. We propose a new image flattening method that uses a vector shift field, instead of a scalar field of vertical shifts, to represent deformation in an image. The method can flatten by vertically shearing or by rotating portions of an image, or by a combination of vertical shear and rotation. Because it is not limited to vertical shearing, the method can flatten in ways more consistent with geologic deformation.

#### 4.2 Introduction

To interpret stratigraphic features in a seismic image, it is helpful to view an *isochron* image—an image of constant geologic time. However, because of structural deformation, the axes of seismic images are rarely aligned with geologic time. Thus, it is necessary to identify isochrons. This can be accomplished by manual picking of horizons, or alternatively, by automatic seismic image flattening.

Most methods for automatic flattening are limited to vertical shearing of images (Lomask *et al.*, 2006; Parks, 2010). These methods can flatten well images in which the geologic

---

<sup>1</sup> Center for Wave Phenomena, Department of Geophysics, Colorado School of Mines

deformation itself contains only vertical shifts, but for images with more complicated deformations, flattening by vertical shearing may significantly distort image features. For example, non-vertical shifts are necessary to flatten horizons that are faulted, folded, or overturned. A flattening method should be consistent with geologic deformation; but methods that are limited to vertical shearing of an image clearly cannot represent all manner of such deformation.

We propose a new method for automatic flattening of seismic images that represents deformation in an image using a shift vector field instead of a scalar field of vertical shifts, and then flattens the image by reversing the deformation. Because the method is not limited to vertical shearing, it can better represent actual geologic deformation, and can thereby minimize non-geologic distortions of image features.

#### 4.2.1 Structure tensors

To flatten image features, we need a measure of their orientation. We use the structure tensor (Fehmers & Höcker, 2003; van Vliet & Verbeek, 1995) to compute normal vectors perpendicular to locally linear features in 2D images or locally planar features in 3D images.

The structure tensor, also called the gradient-squared tensor, is a smoothed outer product of image gradients. In 2D, the structure tensor  $\mathbf{T}$  for a single image sample is a  $2 \times 2$  symmetric positive-semidefinite matrix:

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{13} \\ t_{13} & t_{33} \end{bmatrix}, \quad (4.1)$$

where  $t_{ij}$  is a smoothed product of image derivatives in the  $x_i$  and  $x_j$  directions.

The eigendecomposition of  $\mathbf{T}$  describes the orientation of features in an image (Fehmers & Höcker, 2003). For a 2D image, the eigendecomposition is

$$\mathbf{T} = \lambda_u \mathbf{u}\mathbf{u}^\top + \lambda_v \mathbf{v}\mathbf{v}^\top, \quad (4.2)$$

where  $\lambda_u$  and  $\lambda_v$  are the eigenvalues corresponding to eigenvectors  $\mathbf{u}$  and  $\mathbf{v}$ , respectively. By convention,  $\lambda_u \geq \lambda_v \geq 0$ .

The eigenvector  $\mathbf{u}$  corresponding to the largest eigenvalue describes the direction of the highest image derivative, and therefore is orthogonal to linear features in an image. In other words, eigenvector  $\mathbf{u}$  is the normal vector. Figure 4.1a shows a subset of the normal vectors computed for a 2D seismic image.

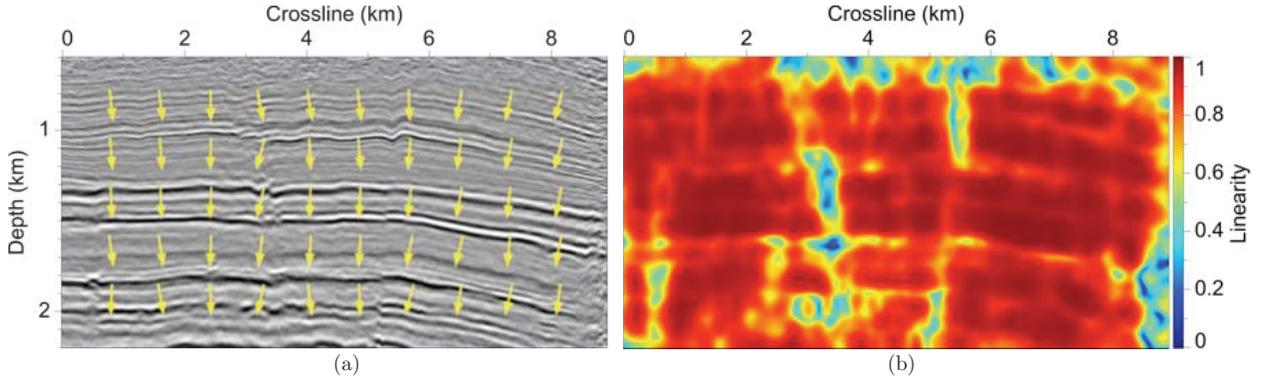


Figure 4.1: Normal vectors (a) and linearity (b) computed for a 2D seismic image.

The eigenvalues  $\lambda_u$  and  $\lambda_v$  provide a measure of the linearity of image features. The linearity  $\lambda_1$  is computed as

$$\lambda_1 = (\lambda_u - \lambda_v) / \lambda_u. \quad (4.3)$$

Note that  $0 \leq \lambda_1 \leq 1$ . For coherent, linear features ( $\lambda_u \gg \lambda_v$ ) linearity approaches unity, whereas for incoherent features ( $\lambda_u = \lambda_v$ ) linearity approaches zero. Figure 4.1b shows linearity computed for the seismic image shown in Figure 4.1a. Notice that in noisy areas (e.g., the lower right corner of the image) and in areas where features are less coherent (e.g., near faults), linearity is closer to zero. When flattening, we use linearity to give more weight to coherent, linear image features.

In 3D, we compute image derivatives in three orthogonal directions. Accordingly, each structure tensor is a  $3 \times 3$  symmetric positive-semidefinite matrix:

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{12} & t_{22} & t_{23} \\ t_{13} & t_{23} & t_{33} \end{bmatrix}, \quad (4.4)$$

with eigendecomposition

$$\mathbf{T} = \lambda_u \mathbf{u}\mathbf{u}^\top + \lambda_v \mathbf{v}\mathbf{v}^\top + \lambda_w \mathbf{w}\mathbf{w}^\top. \quad (4.5)$$

For 3D image flattening, we are interested in planar, rather than linear, features. Planarity is given by

$$\lambda_2 = (\lambda_u - \lambda_v)/\lambda_u. \quad (4.6)$$

As for linearity in 2D, we use planarity to give more weight to coherent, planar image features when flattening in 3D.

### 4.3 Image flattening

Let  $f(\mathbf{x})$  denote an input image and let  $g(\mathbf{u})$  denote the flattened image, where  $\mathbf{x}$  is a point in present-day coordinates and  $\mathbf{u}$  a point in the flattened coordinates. In 2D,  $\mathbf{x} = (x_1, x_3)$  and  $\mathbf{u} = (u_1, u_3)$ ; in 3D,  $\mathbf{x} = (x_1, x_2, x_3)$  and  $\mathbf{u} = (u_1, u_2, u_3)$ . Subscripts 1 and 2 denote horizontal axes, while subscript 3 denotes the vertical depth axis.

To flatten an image, we need a mapping  $\mathbf{x}(\mathbf{u})$  that specifies the present-day location  $\mathbf{x}$  of any point  $\mathbf{u}$  in the flattened space. Given  $\mathbf{x}(\mathbf{u})$ , we may compute the flattened image by

$$g(\mathbf{u}) = f(\mathbf{x}(\mathbf{u})). \quad (4.7)$$

In the flattened space, constant  $u_3$  corresponds to constant geologic time. Thus, the corresponding surface  $\mathbf{x}(\mathbf{u})$  for constant  $u_3$  is a geologic horizon.

Consider a point  $\mathbf{x} = \mathbf{x}(\mathbf{u})$  located on a geologic horizon, located in the infinitesimal neighborhood of a point  $\mathbf{x}_0 = \mathbf{x}(\mathbf{u}_0)$  on the same horizon. The first-order Taylor series approximation for  $\mathbf{x}$  is

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{J}(\mathbf{u} - \mathbf{u}_0), \quad (4.8)$$

where  $\mathbf{J} = \partial\mathbf{x}/\partial\mathbf{u}$  is the Jacobian of the transformation  $\mathbf{x}(\mathbf{u})$ . Let  $\mathbf{n}$  denote the unit normal vector at  $\mathbf{x}_0$ , and let  $\mathbf{m}$  denote the corresponding normal vector at  $\mathbf{u}_0$ . Then, because

$$\mathbf{n}^\top(\mathbf{x} - \mathbf{x}_0) = \mathbf{n}^\top \mathbf{J}(\mathbf{u} - \mathbf{u}_0) = \mathbf{m}^\top(\mathbf{u} - \mathbf{u}_0) = 0, \quad (4.9)$$

we have

$$\mathbf{J}^T \mathbf{n} = \mathbf{m}. \quad (4.10)$$

Because the normal vector in the flattened space must point downward for the image to be flat, all components of  $\mathbf{m}$  except for  $m_3$  must be zero. Note that we compute normal vectors  $\mathbf{n}$  from the input image, so that  $\mathbf{n} = \mathbf{n}(\mathbf{x})$ , not  $\mathbf{n}(\mathbf{u})$ .

### 4.3.1 Flattening in 2D

The mapping  $\mathbf{x}(\mathbf{u})$  may be written in terms of a shift vector field  $\mathbf{r}(\mathbf{u})$ :

$$\mathbf{x}(\mathbf{u}) = \mathbf{u} - \mathbf{r}(\mathbf{u}). \quad (4.11)$$

The Jacobian for this mapping in 2D is

$$\mathbf{J} = \begin{bmatrix} \partial x_1 / \partial u_1 & \partial x_1 / \partial u_3 \\ \partial x_3 / \partial u_1 & \partial x_3 / \partial u_3 \end{bmatrix}, \quad (4.12)$$

which, in terms of the shift field  $\mathbf{r}(\mathbf{u})$ , is

$$\mathbf{J} = \begin{bmatrix} 1 - r_{11} & -r_{13} \\ -r_{31} & 1 - r_{33} \end{bmatrix}, \quad (4.13)$$

where  $r_{ij} \equiv \partial r_i / \partial u_j$ . From equation 4.10, we have

$$n_1(1 - r_{11}) - n_3 r_{31} = 0 \quad (4.14)$$

$$n_3(1 - r_{33}) - n_1 r_{13} = m_3 \quad (4.15)$$

where  $m_3$  is the vertical component of the normal vector after flattening, and is related to the type of deformation in the shift vector field  $\mathbf{r}(\mathbf{u})$ . Equation 4.14 can be considered the flattening equation—if the shift vector field  $\mathbf{r}(\mathbf{u})$  satisfies this equation for  $n_1$  and  $n_3$  computed from an image, then applying the shifts  $\mathbf{r}(\mathbf{u})$  will flatten the image. We write equation 4.13 as

$$\mathbf{J} = \mathbf{I} - \mathbf{R}, \quad (4.16)$$

where  $\mathbf{I}$  is a  $2 \times 2$  identity matrix, and  $\mathbf{R}$  is a matrix of partial derivatives of the shift vector field  $\mathbf{r}(\mathbf{u})$ :

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{13} \\ r_{31} & r_{33} \end{bmatrix}. \quad (4.17)$$

Solving equation 4.16 for  $\mathbf{R}$  gives

$$\mathbf{R} = \mathbf{I} - \mathbf{J}. \quad (4.18)$$

To find  $\mathbf{R}$ , we need an appropriate Jacobian that satisfies equation 4.10.

We consider separately two different methods: flattening by vertical shearing, and flattening by rotation. The Jacobian for vertical shearing in 2D is

$$\mathbf{J}_v = \begin{bmatrix} 1 & 0 \\ -n_1/n_3 & 1 \end{bmatrix}, \quad (4.19)$$

where  $n_1$  and  $n_3$  are the components of the unit normal vector  $\mathbf{n}$ , and the subscript  $v$  indicates vertical shearing. For rotation, the Jacobian is

$$\mathbf{J}_r = \begin{bmatrix} n_3 & n_1 \\ -n_1 & n_3 \end{bmatrix}, \quad (4.20)$$

where the subscript  $r$  indicates rotation. Figure 4.2 shows the mapping produced by these Jacobian matrices. In the figure,  $\mathbf{n}$  is the normal vector in present-day space, and  $\mathbf{j}_1$  and  $\mathbf{j}_3$  are, respectively, the first and second column of the Jacobian. Figure 4.2b shows the rotation Jacobian  $\mathbf{J}_r$ , while Figure 4.2c shows the vertical shear Jacobian  $\mathbf{J}_v$ . Both rotation and vertical shear are area-preserving transformations; thus, a unit area in flattened space maps to either a rotated or sheared unit area in present-day space. Note that thickness measured in the direction of the normal vector is preserved only in the case of rotation.

From equation 4.18 with  $\mathbf{J} = \mathbf{J}_v$ , we obtain the matrix of partial derivatives of the shift vector field for vertical shearing:

$$\mathbf{R}_v = \begin{bmatrix} 0 & 0 \\ n_1/n_3 & 0 \end{bmatrix}. \quad (4.21)$$

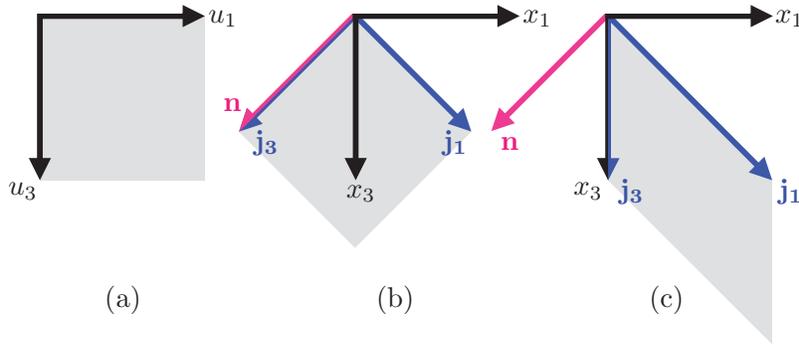


Figure 4.2: A unit area (a) in flattened space maps to a rotated unit area (b) or a sheared unit area (c) in present-day space.

Similarly, equation 4.18 with  $\mathbf{J} = \mathbf{J}_r$  gives the partial derivative matrix for rotation:

$$\mathbf{R}_r = \begin{bmatrix} 1 - n_3 & -n_1 \\ n_1 & 1 - n_3 \end{bmatrix}. \quad (4.22)$$

Matrices  $\mathbf{R}_v$  and  $\mathbf{R}_r$  describe the shift vector fields for pure vertical shear or for pure rotation; but to represent more complex geologic deformations, we must combine both methods. One simple combination is

$$\mathbf{R}_c = (1 - \alpha) \mathbf{R}_v + \alpha \mathbf{R}_r. \quad (4.23)$$

In terms of the components of the normal vector and the parameter  $\alpha$ ,

$$\mathbf{R}_c = \begin{bmatrix} \alpha(1 - n_3) & -\alpha n_1 \\ \alpha n_1 + (1 - \alpha)n_1/n_3 & \alpha(1 - n_3) \end{bmatrix}. \quad (4.24)$$

If we know (or assume) the type of geologic deformation that occurred, then we may choose the corresponding  $\alpha \in [0, 1]$ . Note that  $\alpha = 0$  produces vertical shearing, while  $\alpha = 1$  produces rotation. We equate equation 4.17 and equation 4.24

$$\mathbf{R} = \mathbf{R}_c, \quad (4.25)$$

to obtain four equations for the partial derivatives of the shift vector field

$$r_{11} = \alpha(1 - n_3) \quad (4.26)$$

$$r_{13} = -\alpha n_1 \quad (4.27)$$

$$r_{31} = \alpha n_1 + (1 - \alpha)n_1/n_3 \quad (4.28)$$

$$r_{33} = \alpha(1 - n_3), \quad (4.29)$$

which we solve for  $\mathbf{r}(\mathbf{u})$  by weighted least-squares. Note from equation 4.14 that  $r_{11}$  and  $r_{31}$  are related to flattening. The remaining two equations for  $r_{13}$  and  $r_{33}$  are related to area preservation, as they determine the length of normal vector  $\mathbf{m}$  in the flattened space. If we weight all four equations equally, the method will attempt to flatten an image while scaling locally the area of the image by the determinant of the Jacobian. For most seismic images, it is not possible to flatten everywhere if we constrain the area. Thus, when solving for  $\mathbf{r}(\mathbf{u})$ , we give more weight to equations 4.26 and 4.28 to ensure flattening.

It is simple to verify that the Jacobian for  $\mathbf{R}_c$  satisfies equation 4.10. The composite Jacobian is

$$\mathbf{J}_c = \mathbf{I} - \mathbf{R}_c, \quad (4.30)$$

and for  $\mathbf{J}_c^T \mathbf{n}$ , we have

$$\begin{bmatrix} 1 - \alpha(1 - n_3) & -\alpha n_1 - (1 - \alpha)n_1/n_3 \\ \alpha n_1 & 1 - \alpha(1 - n_3) \end{bmatrix} \begin{bmatrix} n_1 \\ n_3 \end{bmatrix} = \begin{bmatrix} 0 \\ \alpha + n_3(1 - \alpha) \end{bmatrix}.$$

As expected, the horizontal component of the normal vector  $\mathbf{m}$  in the flattened space is zero. Note that for  $\alpha = 1$ ,  $\mathbf{m}$  is also a unit vector, which indicates that flattening by rotation preserves the thickness (measured perpendicular to bedding) of sedimentary layers. This is not the case for flattening by vertical shear ( $\alpha = 0$ ).

One problem with equation 4.25 is that that shifts  $\mathbf{r}(\mathbf{u})$  are a function of  $\mathbf{u}$ , so that  $\mathbf{R} = \mathbf{R}(\mathbf{u})$ ; but the normal vectors  $\mathbf{n}(\mathbf{x})$  are functions of  $\mathbf{x}$ , so that  $\mathbf{R}_c = \mathbf{R}_c(\mathbf{x})$ . Therefore, the equation we must solve is

$$\mathbf{R}(\mathbf{u}) = \mathbf{R}_c(\mathbf{x}) = \mathbf{R}_c(\mathbf{u} - \mathbf{r}(\mathbf{u})). \quad (4.31)$$

This equation is nonlinear, because the right-hand-side depends on the solution  $\mathbf{r}(\mathbf{u})$ . In practice, we use a simple fixed-point iteration to handle this (generally weak) nonlinearity. We begin with  $\mathbf{r}(\mathbf{u}) = 0$ , compute  $\mathbf{R}_c(\mathbf{u})$ , solve for  $\mathbf{r}(\mathbf{u})$ , compute  $\mathbf{R}_c(\mathbf{u} - \mathbf{r}(\mathbf{u}))$ , solve for  $\mathbf{r}(\mathbf{u})$ , and so on until converged. Convergence is fast when the normal vectors vary slowly, i.e., when  $\mathbf{n}(\mathbf{u}) \approx \mathbf{n}(\mathbf{x})$ .

Figure 4.3 and Figure 4.4 show, respectively, examples of flattening by vertical shear and flattening by rotation for a synthetic seismic image. Notice that for the vertical shear case, the horizontal component of the shift vector field  $\mathbf{r}(\mathbf{u})$  shown in Figure 4.3c is zero, which indicates traces are being shifted only vertically. Notice also that thickness measured in the direction normal to the layering is preserved only in the case of rotation (Figure 4.4b).

Figure 4.5 and Figure 4.6 show examples of flattening by vertical shear and flattening by rotation for a 2D slice from a 3D seismic image of Teapot Dome. Note that the images for these examples are stretched vertically, and that if we were to plot the input seismic image with a 1:1 aspect ratio, we would see that the image is nearly flat. Because flattening by vertical shear and flattening by rotation produce identical results for an already flat image, it is not surprising that Figure 4.5 and Figure 4.6 are similar.

We may exaggerate the structure in a seismic image by increasing the sampling interval in the depth direction. Figure 4.7 shows an example of flattening by rotation for the same 2D seismic image shown in Figure 4.5a and Figure 4.6a, but with the depth sampling interval increased from 4 m to 25 m. Notice that the horizontal components  $r_1(\mathbf{u})$  of the shift vector field  $\mathbf{r}(\mathbf{u})$  shown in Figure 4.7c are larger than those shown in Figure 4.6c, which are nearly zero.

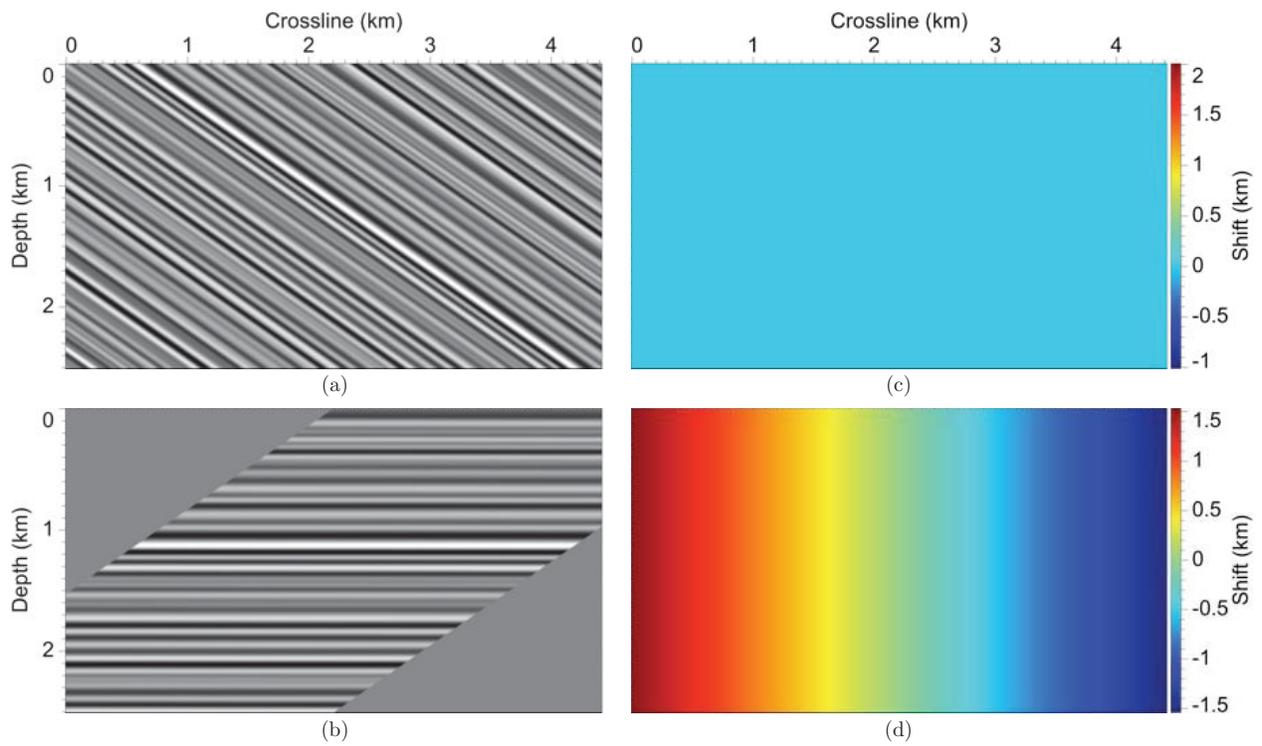


Figure 4.3: A synthetic image (a) flattened by vertical shear (b) with the horizontal (c) and vertical (d) components of the shift vector field  $\mathbf{r}(\mathbf{u})$ .

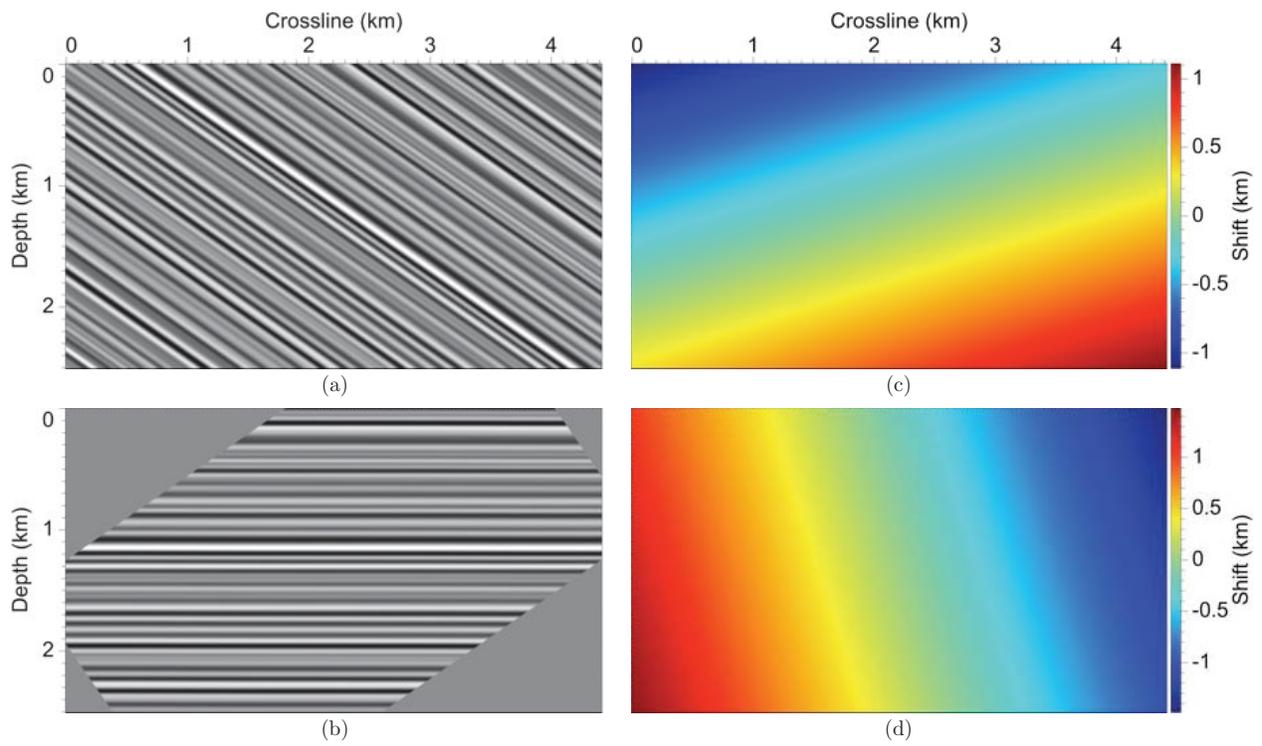


Figure 4.4: A synthetic image (a) flattened by rotation (b) with the horizontal (c) and vertical (d) components of the shift vector field  $\mathbf{r}(\mathbf{u})$ .

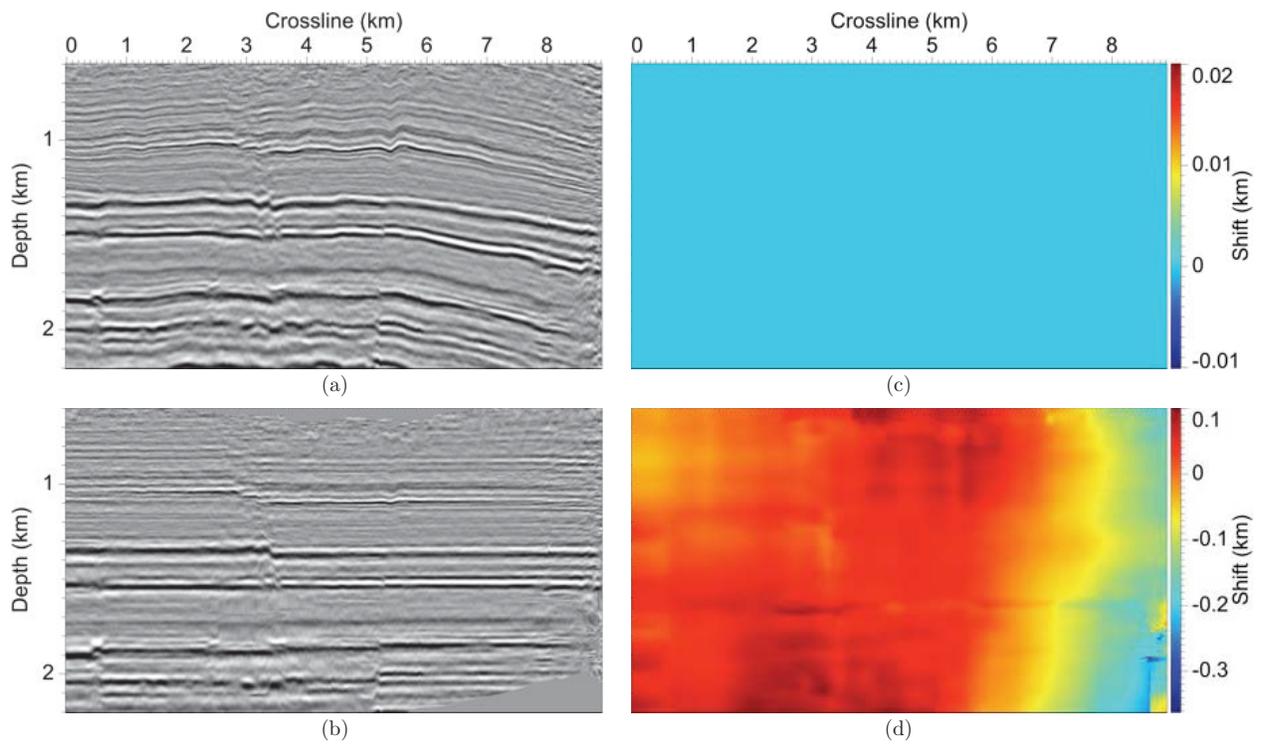


Figure 4.5: A seismic image (a) flattened by vertical shear (b) with the horizontal (c) and vertical (d) components of the shift vector field  $\mathbf{r}(\mathbf{u})$ .

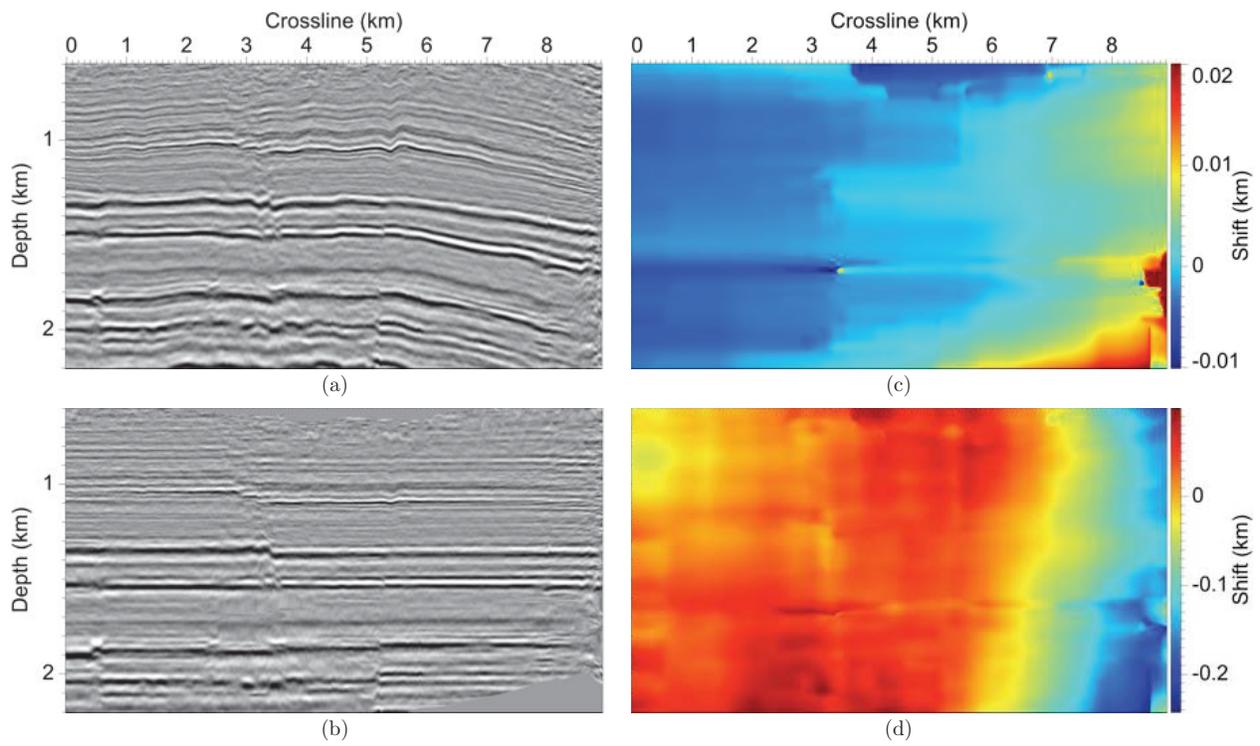


Figure 4.6: A seismic image (a) flattened by rotation (b) with the horizontal (c) and vertical (d) components of the shift vector field  $\mathbf{r}(\mathbf{u})$ .

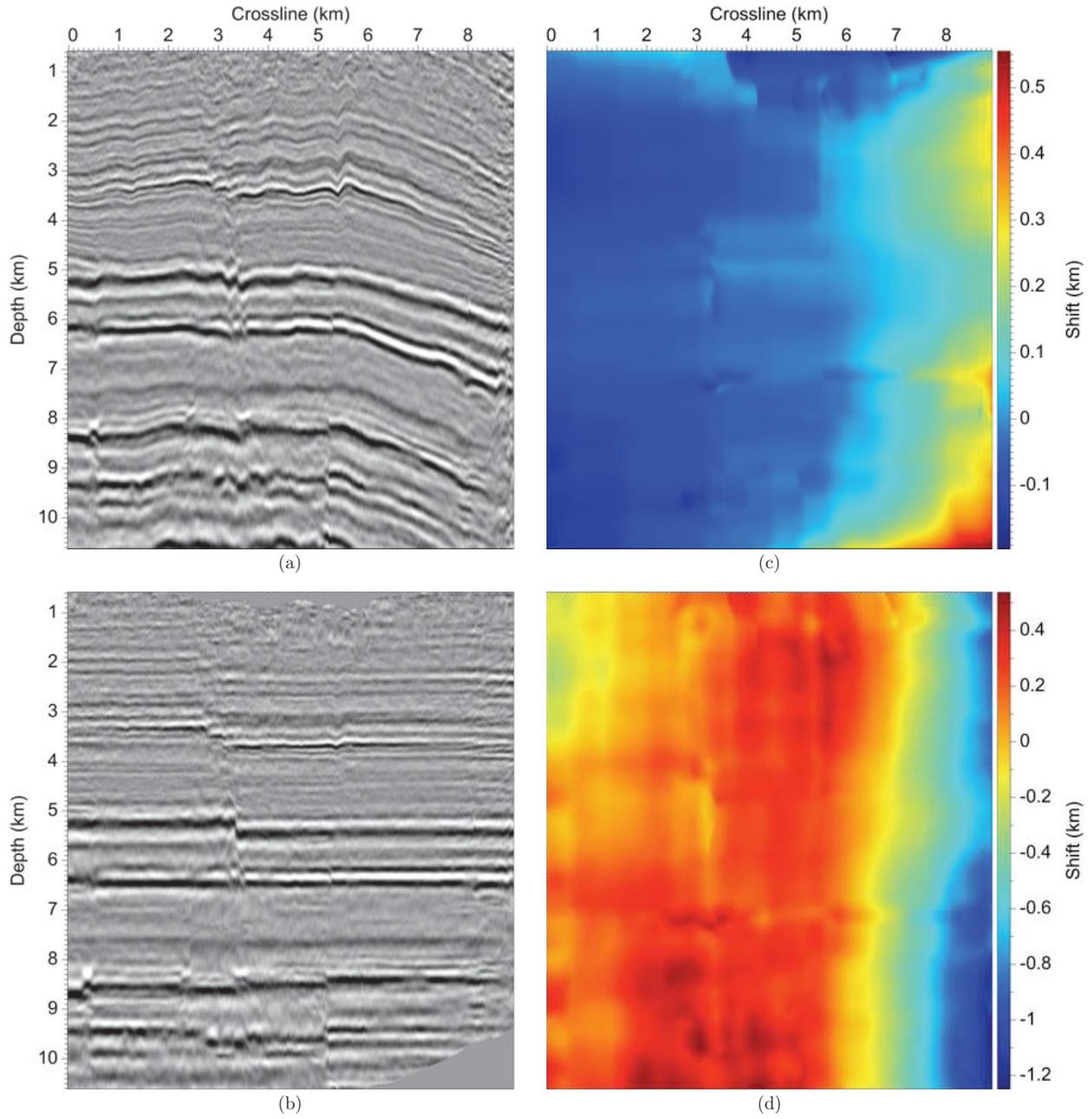


Figure 4.7: A seismic image (a) flattened by rotation (b) with the horizontal (c) and vertical (d) components of the shift vector field  $\mathbf{r}(\mathbf{u})$ .

### 4.3.2 Flattening in 3D

The extension of our flattening method to 3D is straightforward. In 3D, the Jacobian matrix for a mapping  $\mathbf{x}(\mathbf{u})$  is

$$\mathbf{J} = \begin{bmatrix} \partial x_1/\partial u_1 & \partial x_1/\partial u_2 & \partial x_1/\partial u_3 \\ \partial x_2/\partial u_1 & \partial x_2/\partial u_2 & \partial x_2/\partial u_3 \\ \partial x_3/\partial u_1 & \partial x_3/\partial u_2 & \partial x_3/\partial u_3 \end{bmatrix}, \quad (4.32)$$

which, from equation 4.11, is equivalent to

$$\mathbf{J} = \begin{bmatrix} 1 - r_{11} & -r_{12} & -r_{13} \\ -r_{21} & 1 - r_{22} & -r_{23} \\ -r_{31} & -r_{32} & 1 - r_{33} \end{bmatrix}. \quad (4.33)$$

where  $r_{ij} \equiv \partial r_i/\partial u_j$ . From equation 4.10, we have

$$n_1(1 - r_{11}) - n_2 r_{21} - n_3 r_{31} = 0 \quad (4.34)$$

$$n_2(1 - r_{22}) - n_1 r_{12} - n_3 r_{32} = 0 \quad (4.35)$$

$$n_3(1 - r_{33}) - n_1 r_{13} - n_2 r_{23} = m_3. \quad (4.36)$$

Equations 4.34 and 4.35 are the flattening equations; given normal vectors computed from an image, applying any shift vector field  $\mathbf{r}(\mathbf{u})$  that satisfies both equations will flatten the image. Equation 4.33 may be written as

$$\mathbf{J} = \mathbf{I} - \mathbf{R}, \quad (4.37)$$

where  $\mathbf{I}$  is a  $3 \times 3$  identity matrix, and  $\mathbf{R}$  is a matrix of partial derivatives of the shift vector field:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (4.38)$$

Solving equation 4.37 for  $\mathbf{R}$ , we obtain

$$\mathbf{R} = \mathbf{I} - \mathbf{J}. \quad (4.39)$$

To find  $\mathbf{R}$ , we need a Jacobian matrix that satisfies equation 4.10.

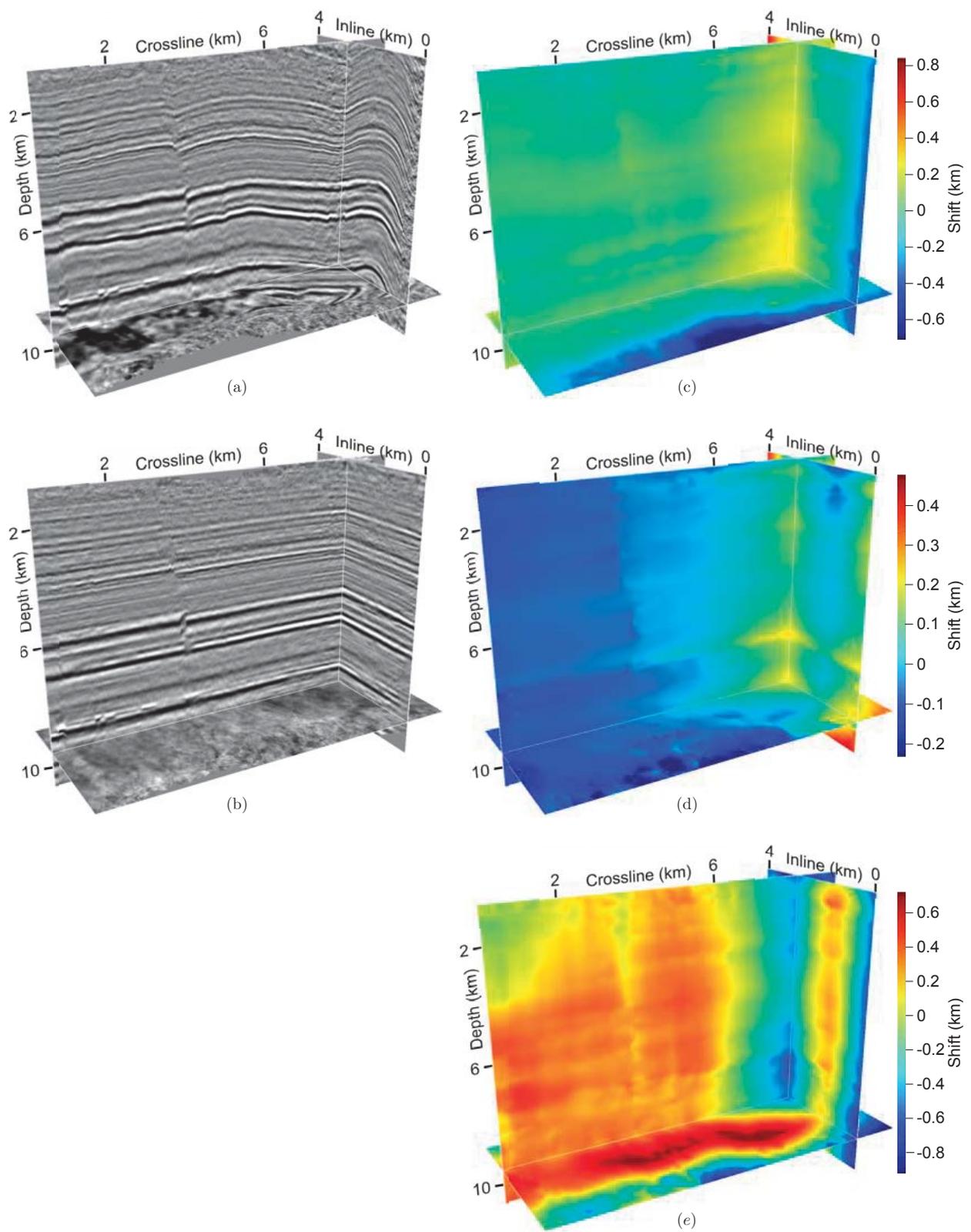


Figure 4.8: A 3D seismic image (a) flattened by rotation (b) with the inline shift  $r_1$  (c), crossline shift  $r_2$  (d), and vertical shift  $r_3$  (e) of the shift vector field  $\mathbf{r}(\mathbf{u})$ .

Again, we consider separately the Jacobian matrix for flattening by vertical shear and flattening by rotation. For vertical shear in 3D, the Jacobian is

$$\mathbf{J}_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -n_1/n_3 & -n_2/n_3 & 1 \end{bmatrix}, \quad (4.40)$$

and for rotation

$$\mathbf{J}_r = \begin{bmatrix} n_3 + n_2^2/(1+n_3) & -n_1n_2/(1+n_3) & n_1 \\ -n_1n_2/(1+n_3) & n_3 + n_1^2/(1+n_3) & n_2 \\ -n_1 & -n_2 & n_3 \end{bmatrix}. \quad (4.41)$$

From equation 4.18 we obtain the corresponding matrix of partial derivatives for vertical shear

$$\mathbf{R}_v = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ n_1/n_3 & n_2/n_3 & 0 \end{bmatrix}, \quad (4.42)$$

and for rotation

$$\mathbf{R}_r = \begin{bmatrix} n_1^2/(1+n_3) & n_1n_2/(1+n_3) & -n_1 \\ n_1n_2/(1+n_3) & n_2^2/(1+n_3) & -n_2 \\ n_1 & n_2 & 1-n_3 \end{bmatrix}. \quad (4.43)$$

We form a composite matrix as a linear combination of  $\mathbf{R}_v$  and  $\mathbf{R}_r$ :

$$\mathbf{R}_c = (1 - \alpha) \mathbf{R}_v + \alpha \mathbf{R}_r. \quad (4.44)$$

Equating equation 4.38 and 4.44 gives nine equations for the partial derivatives of the shift field:

$$r_{11} = \alpha n_1^2/(1+n_3) \quad (4.45)$$

$$r_{12} = \alpha n_1 n_2/(1+n_3) \quad (4.46)$$

$$r_{13} = -\alpha n_1 \quad (4.47)$$

$$r_{21} = \alpha n_1 n_2/(1+n_3) \quad (4.48)$$

$$r_{22} = \alpha n_2^2/(1+n_3) \quad (4.49)$$

$$r_{23} = -\alpha n_2 \quad (4.50)$$

$$r_{31} = \alpha n_1 + (1-\alpha)n_1/n_3 \quad (4.51)$$

$$r_{32} = \alpha n_2 + (1-\alpha)n_2/n_3 \quad (4.52)$$

$$r_{33} = \alpha(1-n_3) \quad (4.53)$$

The flattening equations 4.34 and 4.35 contain terms  $r_{11}$ ,  $r_{12}$ ,  $r_{21}$ ,  $r_{22}$ ,  $r_{31}$ , and  $r_{32}$ . Thus, when solving for  $\mathbf{r}(\mathbf{u})$ , we give more weight to the corresponding equations 4.45, 4.46, 4.48, 4.49, 4.51, and 4.52. The remaining equations for  $r_{13}$ ,  $r_{23}$  and  $r_{33}$  determine the length of the normal vector  $\mathbf{m}$ , and therefore are related to volume preservation. These equations are usually given less weight in order to ensure flattening. Figure 4.8 shows an example of flattening by rotation ( $\alpha = 1$ ) for an image of Teapot Dome, with the depth sampling interval increased from 4 m to 25 m to exaggerate the structure in the image.

#### 4.4 Inverse shift vectors?

For image flattening using vector shifts, we considered the mapping

$$\mathbf{x}(\mathbf{u}) = \mathbf{u} - \mathbf{r}(\mathbf{u}), \quad (4.54)$$

which describes the deformation of a point  $\mathbf{u}$  in the flattened space. Alternatively, we may consider the inverse mapping

$$\mathbf{u}(\mathbf{x}) = \mathbf{x} + \mathbf{s}(\mathbf{x}), \quad (4.55)$$

where  $\mathbf{s}(\mathbf{x})$  is a shift vector field that satisfies

$$\mathbf{s}(\mathbf{x}) = \mathbf{r}(\mathbf{u}) = \mathbf{r}(\mathbf{x} + \mathbf{s}(\mathbf{x})). \quad (4.56)$$

One advantage of using  $\mathbf{s}(\mathbf{x})$  rather than  $\mathbf{r}(\mathbf{u})$  is that the normal vectors  $\mathbf{n}$  are functions of  $\mathbf{x}$ . Thus, it may be possible to obtain a linear set of equations for  $\mathbf{s}(\mathbf{x})$  that can be solved without the fixed-point iterations required when solving for  $\mathbf{r}(\mathbf{u})$ . In fact, this is the approach used by Parks (2010) to solve for a scalar shift field for vertical shearing only.

Unfortunately, it is difficult to flatten while preserving this linearity when solving for a more general vector shift field. To see this, consider flattening in 3D using the inverse shift vectors  $\mathbf{s}(\mathbf{x})$ . Analogous to equation 4.8, we write the first-order Taylor series approximation for  $\mathbf{u}$  as

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{K}(\mathbf{x} - \mathbf{x}_0), \quad (4.57)$$

where  $\mathbf{K}$  is the Jacobian of the transformation  $\mathbf{u}(\mathbf{x})$ . From equation 4.55, we have

$$\mathbf{K} = \begin{bmatrix} 1 + s_{11} & s_{12} & s_{13} \\ s_{21} & 1 + s_{22} & s_{23} \\ s_{31} & s_{32} & 1 + s_{33} \end{bmatrix} = \mathbf{I} + \mathbf{S}. \quad (4.58)$$

where  $s_{ij} \equiv \partial s_i / \partial x_j$ . The normal vectors before and after flattening are related as

$$\mathbf{n} = \mathbf{K}^T \mathbf{m}, \quad (4.59)$$

which gives

$$\mathbf{K}^{-T} \mathbf{n} = \mathbf{m}. \quad (4.60)$$

Note that the inverse of the Jacobian of the transformation  $\mathbf{u}(\mathbf{x})$  is equal to the inverse of the Jacobian of  $\mathbf{x}(\mathbf{u})$ :

$$\mathbf{K}^{-1} = \mathbf{J}, \quad (4.61)$$

and that equation 4.60 is equivalent to equation 4.10, except that it is written in terms of the shift vectors  $\mathbf{s}(\mathbf{x})$  instead of  $\mathbf{r}(\mathbf{u})$ . For a flattened image, all components of the vector  $\mathbf{m}$  except for  $m_3$  must be zero. Thus to form flattening equations, we need the inverse of the Jacobian  $\mathbf{K}$ :

$$\mathbf{K}^{-1} = \frac{1}{\det \mathbf{K}} \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}, \quad (4.62)$$

where

$$k_{11} = (1 + s_{22})(1 + s_{33}) - s_{23}s_{32} \quad (4.63)$$

$$k_{12} = s_{13}s_{32} - s_{12}(1 + s_{33}) \quad (4.64)$$

$$k_{13} = s_{12}s_{23} - s_{13}(1 + s_{22}) \quad (4.65)$$

$$k_{21} = s_{23}s_{31} - s_{21}(1 + s_{33}) \quad (4.66)$$

$$k_{22} = (1 + s_{11})(1 + s_{33}) - s_{13}s_{31} \quad (4.67)$$

$$k_{23} = s_{13}s_{21} - s_{23}(1 + s_{11}) \quad (4.68)$$

$$k_{31} = s_{21}s_{32} - s_{31}(1 + s_{22}) \quad (4.69)$$

$$k_{32} = s_{12}s_{31} - s_{32}(1 + s_{11}) \quad (4.70)$$

$$k_{33} = (1 + s_{11})(1 + s_{22}) - s_{12}s_{21}. \quad (4.71)$$

From equation 4.60, the flattening equations corresponding to the zero-valued horizontal components of  $\mathbf{m}$  are

$$k_{11}n_1 + k_{12}n_2 + k_{13}n_3 = 0 \quad (4.72)$$

$$k_{21}n_1 + k_{22}n_2 + k_{23}n_3 = 0. \quad (4.73)$$

Unlike flattening equations 4.34 and 4.35 for  $\mathbf{r}(\mathbf{u})$ , which contained only six elements of the matrix  $\mathbf{R}$ , these equations contain all nine elements of the matrix  $\mathbf{S}$ . So to flatten, we need to satisfy all nine equations for the partial derivatives  $s_{ij}$ . Recall that if we weight equally all nine equations, then the method will attempt to both flatten and preserve volume. For most images, it is not possible to flatten and preserve volume everywhere.

By solving for shifts  $\mathbf{r}(\mathbf{u})$ , we avoided this problem by simply weighting the flattening equations. For the inverse shifts  $\mathbf{s}(\mathbf{x})$ , we cannot separate the flattening equations from the volume preserving equations. For this reason, we solve for shifts  $\mathbf{r}(\mathbf{u})$  instead.

## 4.5 Conclusion

The flattening method described in this report is versatile, in that it can flatten an image by vertically shearing the image, by rotating portions of the image, or by a combination of vertical shear and rotation. Because the method solves for a shift vector field, rather than a scalar field of vertical shifts, it is not constrained to flatten by vertical shearing only. A shift vector field can more accurately represent geologic deformation, and can be used to flatten images while minimizing non-geologic distortions of image features.

However, in order to flatten an image according to the true geologic deformation, our method currently requires that we specify the type of deformation that occurred. In practice, it may be difficult to determine the true deformation, even when provided with relevant geologic information. But this shortcoming suggests an interesting research question: can we estimate the type of deformation from the image itself?

One potentially useful measure of the correctness (relative to the true geologic deformation) of a flattening method is the determinant of the Jacobian matrix. In 2D, the

determinant gives the ratio between an infinitesimal area in present-day space and the corresponding area in flattened space. In 3D, the determinant gives the ratio between infinitesimal volumes. Thus, if the determinant is unity, then area or volume has been preserved while flattening. Assuming that the true deformation is area- or volume-preserving, we might use the determinant of the Jacobian to quantify how well a flattening method reflects true geologic deformation.

For now we must either assume, or we must use geologic information to estimate, the type of deformation. In any case, for most images, geologic deformation cannot be described with only a scalar field of vertical shifts. Solving for a vector shift field as described in this report is a simple way to flatten images with non-vertical deformations.

#### **4.6 Acknowledgements**

Thanks to the Rocky Mountain Oilfield Test Center for providing the 3D seismic image of Teapot Dome, and thanks to Transform Software for providing the time-to-depth conversion of that image. Thanks also to Derek Parks and Ran Xuan, whose prior work on seismic image flattening was a helpful starting point for this research. This research was supported by the sponsors of the Center for Wave Phenomena at the Colorado School of Mines.

## CHAPTER 5

### UNFAULTING AND UNFOLDING 3D SEISMIC IMAGES

A paper published in *Geophysics*, 2014, 78 (4), O45-O56

Simon Luo<sup>1</sup> and Dave Hale<sup>1</sup>

#### 5.1 Summary

Identifying and extracting geologic horizons is useful for interpretation of stratigraphic features as well as analysis of structural deformation. To extract horizons from a seismic image, we develop methods for automatically unfauling and unfolding an image to restore all horizons to an undeformed, horizontal state. First, using fault surfaces and dip-separation vectors estimated from an image, we interpolate dip-separation vectors at locations between fault surfaces, and then use the interpolated dip-separation vectors to unfaul an image. Then, using a method for automatic seismic image flattening, we unfold the unfaulked image to obtain a new image in which sedimentary layering is horizontal and also aligned across faults. From this unfaulked and unfolded image, we can automatically extract geologic horizons.

#### 5.2 Introduction

Extracting isochronal geologic surfaces—geologic horizons of the same age—is a common problem in geophysics and geology. Such horizons are useful for interpretation of stratigraphic features and analysis of structural deformation, as well as interpolation and correlation of subsurface properties. Here we consider geologic horizons deposited across much of the area of the seismic survey, such as maximum flooding or other high-stand surfaces. If such surfaces are sequentially subjected to faulting and folding, it is necessary to quantify this deformation in order to extract geologic horizons.

---

<sup>1</sup> Center for Wave Phenomena, Department of Geophysics, Colorado School of Mines

Perhaps the most straightforward way to extract a geologic horizon is by manual picking. Manual picking is often used in conjunction with autotracking methods (e.g., Howard, 1990), which track seismic events by following local extrema or zero crossings in amplitude in a seismic image. Most autotracking methods are not fully automatic, and an experienced interpreter is often required, for example, to identify seed points of coherent events to track, or to correlate events across faults. An obvious disadvantage of manual interpretation is that the process can be slow, because human interaction is required. The advantage, however, is that an experienced interpreter can pick horizons in areas in which a fully automatic method might have difficulty. Such areas could arise from a combination of geological (e.g., faults, unconformities, and complex stratigraphy) and geophysical (e.g., imaging and processing artifacts, noise, and multiples) complications (R. Howard, personal communication).

Alternatives to horizon autotracking methods include volume interpretation methods (e.g., Stark, 1996), which, rather than tracking single events, process simultaneously an entire seismic volume. Automatic seismic image flattening (Lomask *et al.*, 2006; Parks, 2010; Stark, 2004) is an example of a volume interpretation method. Automatic seismic image flattening could potentially identify all horizons in an image, but the method is unable to match horizons across faults unless additional information (e.g., fault slip) is provided. Moreover, most automatic flattening methods are limited to only vertical shearing of an image, but images with non-vertical faults such as the one shown in Figure 5.2(a) clearly cannot be flattened by vertical shear alone.

A fully automatic method (e.g., Tnacheri & Bearnth, 2007) for extracting geologic horizons is ideal. Toward this end, we propose an automatic method that can be used to extract all geologic horizons in an image, consisting of two steps: image unfaulting followed by image unfolding (i.e., image flattening). To unfault a seismic image, we first use the method described by Hale (2013b) to estimate fault locations and dip-separation vectors, the displacement vectors along the dip direction of a fault surface. Then, after interpolating estimated dip-separation vectors at locations between faults, we unfault the image. To un-

fold an unfaulted image, we use non-vertical image flattening (e.g., Luo & Hale, 2011). By unfauling and then unfolding an image, we obtain an image in which a surface of constant relative geologic time maps to a geologic horizon.

### List of Symbols

$\mathbf{x}$  = Coordinates in present-day space.

$\mathbf{w}$  = Coordinates in unfaulted space.

$\mathbf{u}$  = Coordinates in unfolded space.

$\tilde{\mathbf{t}}(\mathbf{w})$  = Dip-separation vectors.

$\mathbf{t}(\mathbf{w})$  = Interpolated dip-separation vectors.

$\tilde{h}(\mathbf{w})$  = Unfaulted image computed using  $\tilde{\mathbf{t}}(\mathbf{w})$ .

$h(\mathbf{w})$  = Unfaulted image computed using  $\mathbf{t}(\mathbf{w})$ .

$\mathbf{n}(\mathbf{x})$  = Normal vectors.

$\mathbf{r}(\mathbf{u})$  = Unfolding shifts.

$\mathbf{s}(\mathbf{u})$  = Combined unfauling and unfolding shifts.

$\mathbf{J}(\mathbf{u})$  = Jacobian matrix corresponding to  $\mathbf{w}(\mathbf{u})$ .

$\mathbf{J}_r(\mathbf{u})$  = Jacobian matrix for rotation.

### 5.3 Image unfauling

To unfaul an image, we must first estimate fault locations and fault slip. For the examples shown in this paper, we use the method described by Hale (2013b) to automatically compute fault surfaces and dip separation, the component of fault slip along the dip direction of a fault, from a 3D seismic image. Although we choose to use Hale’s (2013b) method, other methods (e.g., Admasu, 2008; Aurnhammer & Tönnies, 2005; Borgos *et al.*, 2003; Liang *et al.*, 2010) could also be used to estimate fault locations and fault slip.

For an image  $f(\mathbf{x})$ , where  $\mathbf{x} = (x_1, x_2, x_3)$  are coordinates in the present-day space, the estimated dip-separation vectors  $\tilde{\mathbf{t}}(\mathbf{w})$ , where  $\mathbf{w} = (w_1, w_2, w_3)$  are coordinates in the

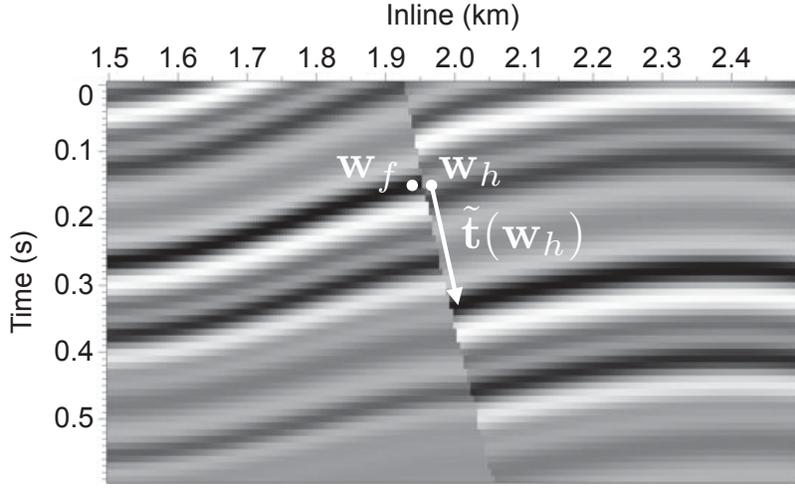


Figure 5.1: A fault dip-separation vector. At  $\mathbf{w}_f$  on the footwall side of the fault, the dip separation is zero. At  $\mathbf{w}_h$  on the hanging wall side of the fault, the dip-separation vector  $\tilde{\mathbf{t}}(\mathbf{w}_h)$  specifies the location of the image sample that aligns with the sample on the footwall side.

unfaulted space, can be used to compute an image

$$\tilde{h}(\mathbf{w}) = f(\mathbf{w} + \tilde{\mathbf{t}}(\mathbf{w})) \quad (5.1)$$

in which seismic events are aligned across faults where  $\tilde{\mathbf{t}}(\mathbf{w})$  is specified. An example of a dip-separation vector for a synthetic 2D seismic image is shown in Figure 5.1. In the figure,  $\mathbf{w}_f$  indicates the location of an image sample on the footwall side of the fault, while  $\mathbf{w}_h$  indicates the location of the corresponding sample on the hanging wall side of the fault. The dip-separation vector  $\tilde{\mathbf{t}}(\mathbf{w}_h)$  specifies the location of the image sample that, once shifted to  $\mathbf{w}_h$  on the hanging wall side, aligns with the image sample at  $\mathbf{w}_f$  on the footwall side. Note from equation 5.1 that events are shifted *only* at locations where the dip separation  $\tilde{\mathbf{t}}(\mathbf{w})$  is specified. Because we estimate dip separation only at locations where we have identified a fault surface, we must interpolate dip-separation vectors at locations between faults to avoid creating new discontinuities in an image when unfauling.

Our convention is that dip-separation vectors  $\tilde{\mathbf{t}}(\mathbf{w})$  specify the separation on the hanging wall side of a fault. Because we will interpolate these dip-separation vectors (e.g., Figure 5.3(a)) between faults, we must also specify dip-separation vectors on the footwall side

of a fault so that the relative separation on opposing sides of a fault do not change after interpolation. Because dip-separation vectors specify the separation on only the hanging wall side of a fault, the dip separation on the footwall side must be zero (see Figure 5.1). For example, Figure 5.2(a) shows a subsection of a 3D seismic image from offshore Netherlands with the vertical component of estimated dip-separation vectors, i.e., the fault throw, overlaid. Notice that on the hanging wall sides of faults, the fault throws are nonzero, while on the footwall sides, the fault throws are zero. Note that Figure 5.2(a) and Figure 5.3(a) show the same fault throws on the hanging wall sides of faults, but only Figure 5.2(a) shows the zero-valued fault throws on the footwall sides of faults.

To interpolate dip-separation vectors at locations between faults, we use blended neighbor interpolation (Hale, 2009a). For Euclidean distances, blended neighbor interpolation is similar to natural neighbor interpolation (Sibson, 1981) and discrete Sibson interpolation (Park *et al.*, 2006). Although we use blended neighbor interpolation, in principle any smooth interpolation would suffice. However, it is important that the interpolation satisfies the interpolation condition, which requires that the interpolant at known locations matches exactly the known values, because we must make sure that interpolation does not change the dip separation estimated at fault locations.

To interpolate a vector field, we interpolate separately each vector component. Figure 5.2(b) shows the blended neighbor interpolation of the estimated fault throws shown in Figure 5.2(a). Using the interpolated dip-separation vectors  $\mathbf{t}(\mathbf{w})$  estimated from an input image  $f(\mathbf{x})$ , the unfaulted image  $h(\mathbf{w})$  is computed as  $h(\mathbf{w}) = f(\mathbf{w} + \mathbf{t}(\mathbf{w}))$ . Figure 5.3(b) shows the unfaulted image computed from the interpolated dip-separation vectors shown in Figure 5.2(b) and the input image shown in Figure 5.2(a). Similarly, Figure 5.6(b) shows the unfaulted image computed from the input image shown in Figure 5.6(a) and the blended neighbor interpolation of the dip-separation vectors whose vertical component is overlaid on the image in Figure 5.6a.

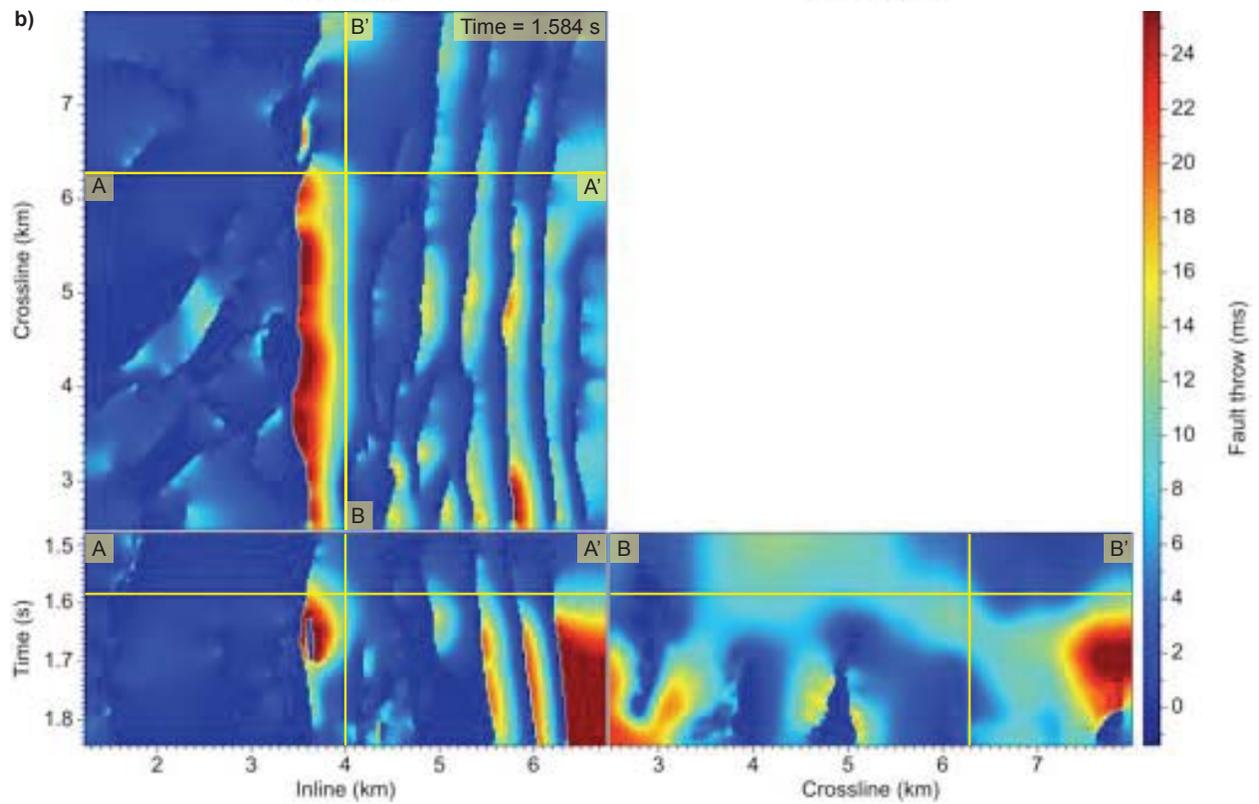
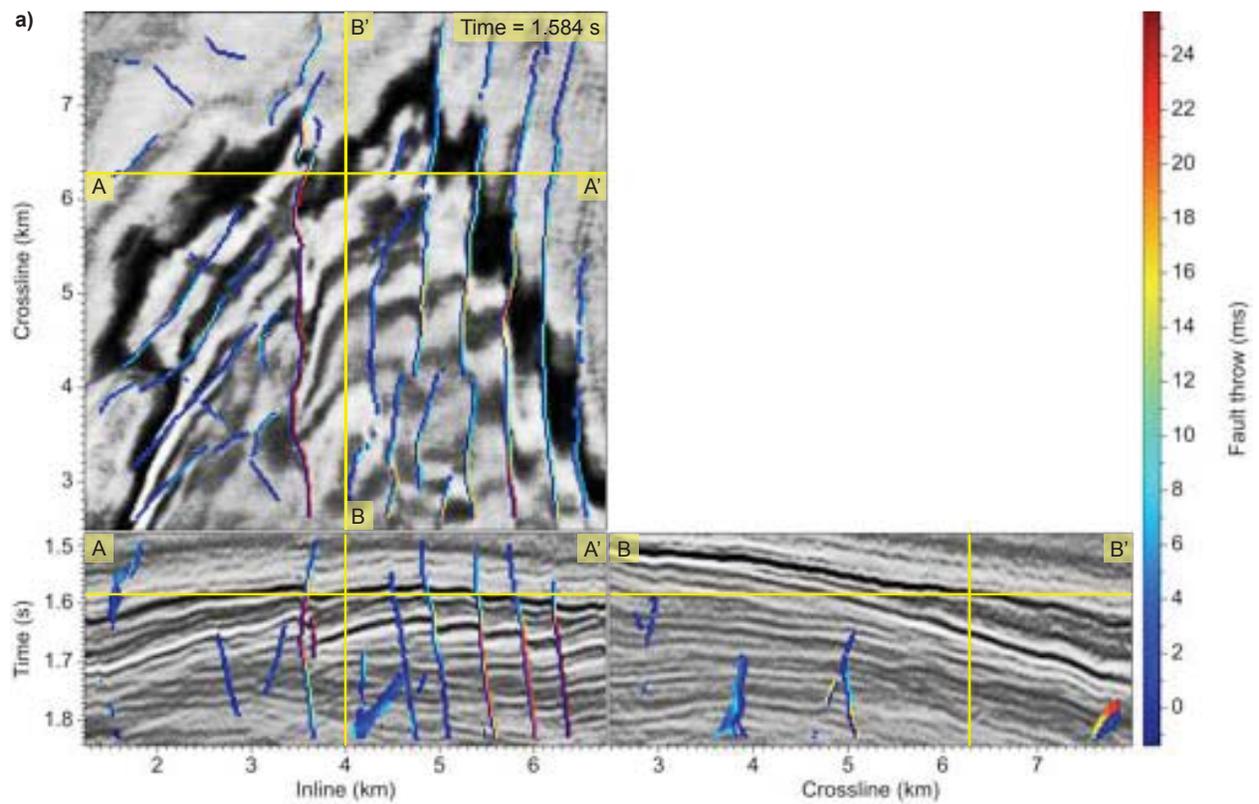


Figure 5.2: A seismic image (a) overlaid with the fault throw, and the blended neighbor interpolation (b) of the fault throw.

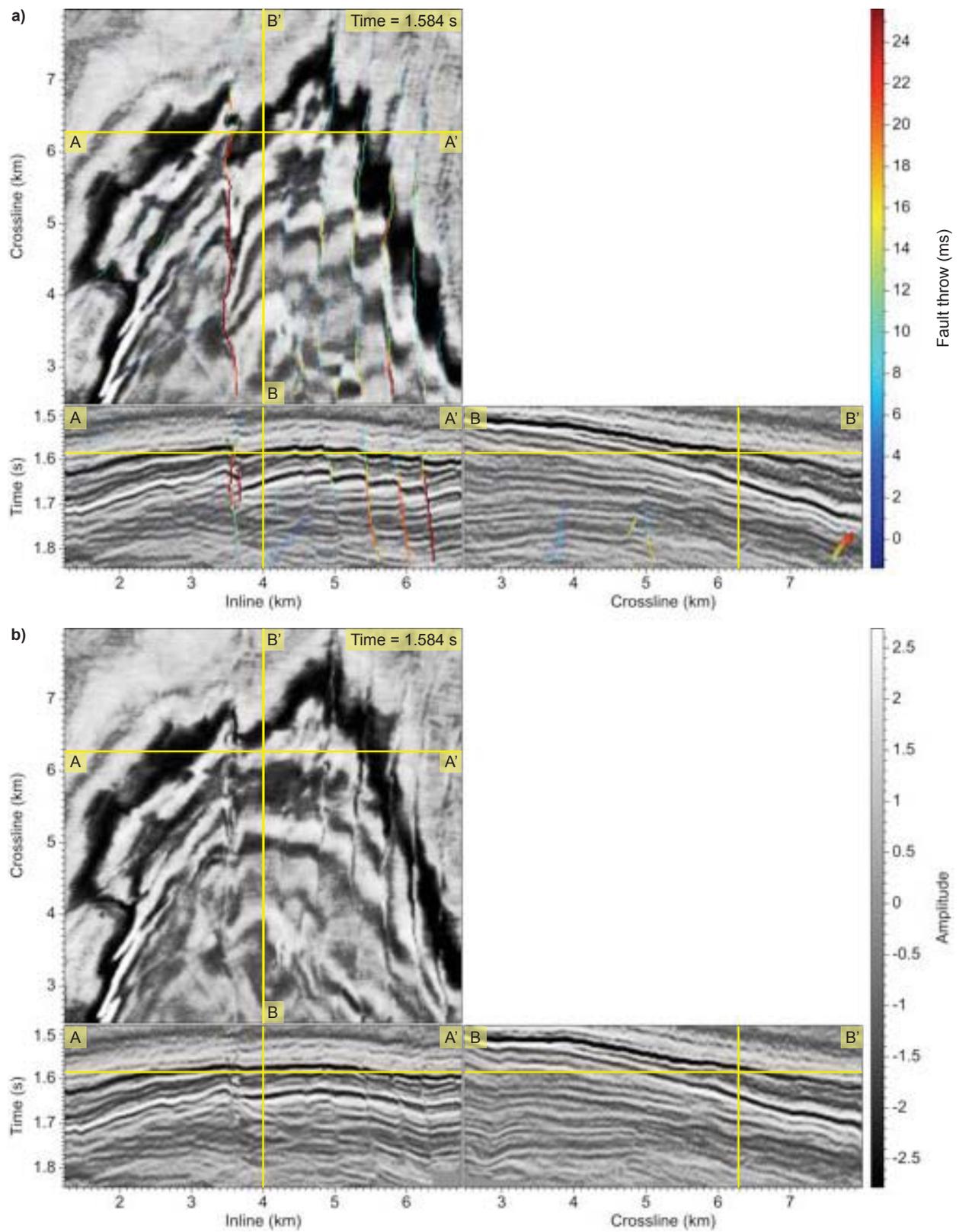


Figure 5.3: A seismic image (a) overlaid with the fault throw, and the unfaulted image (b).

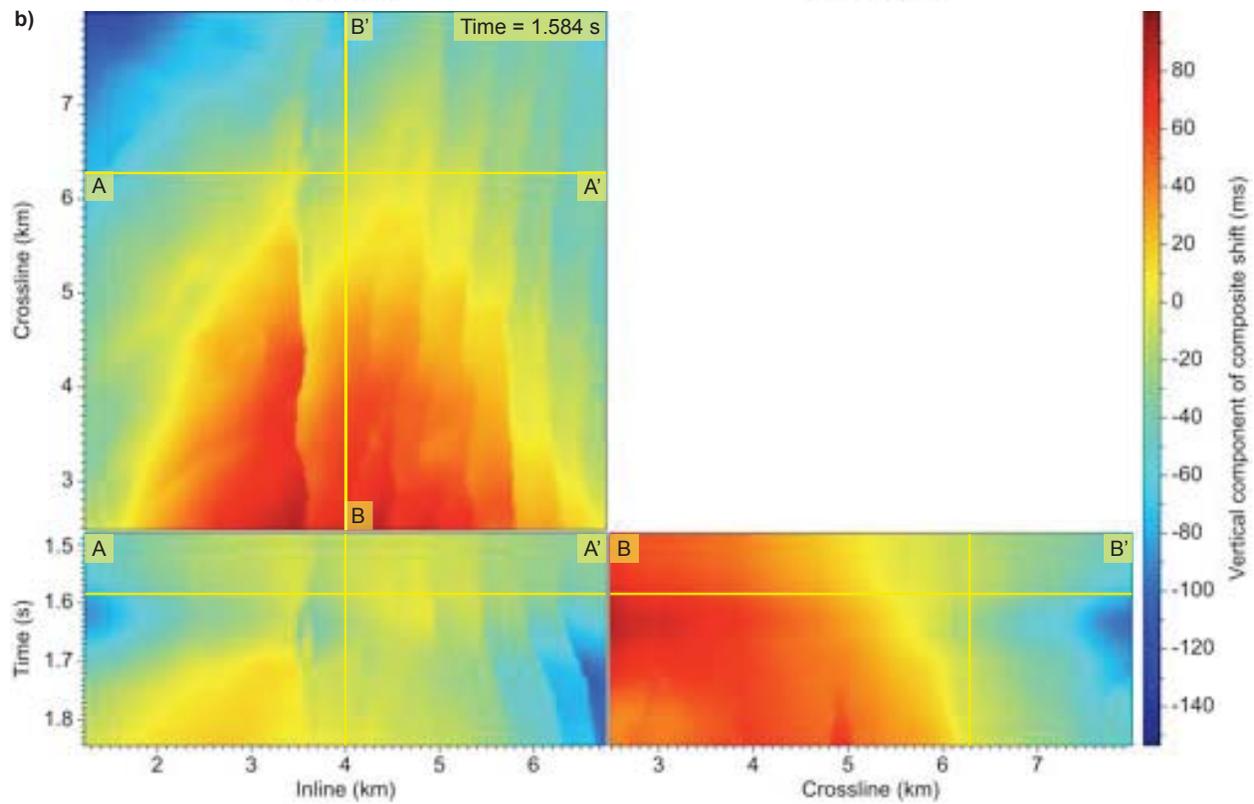
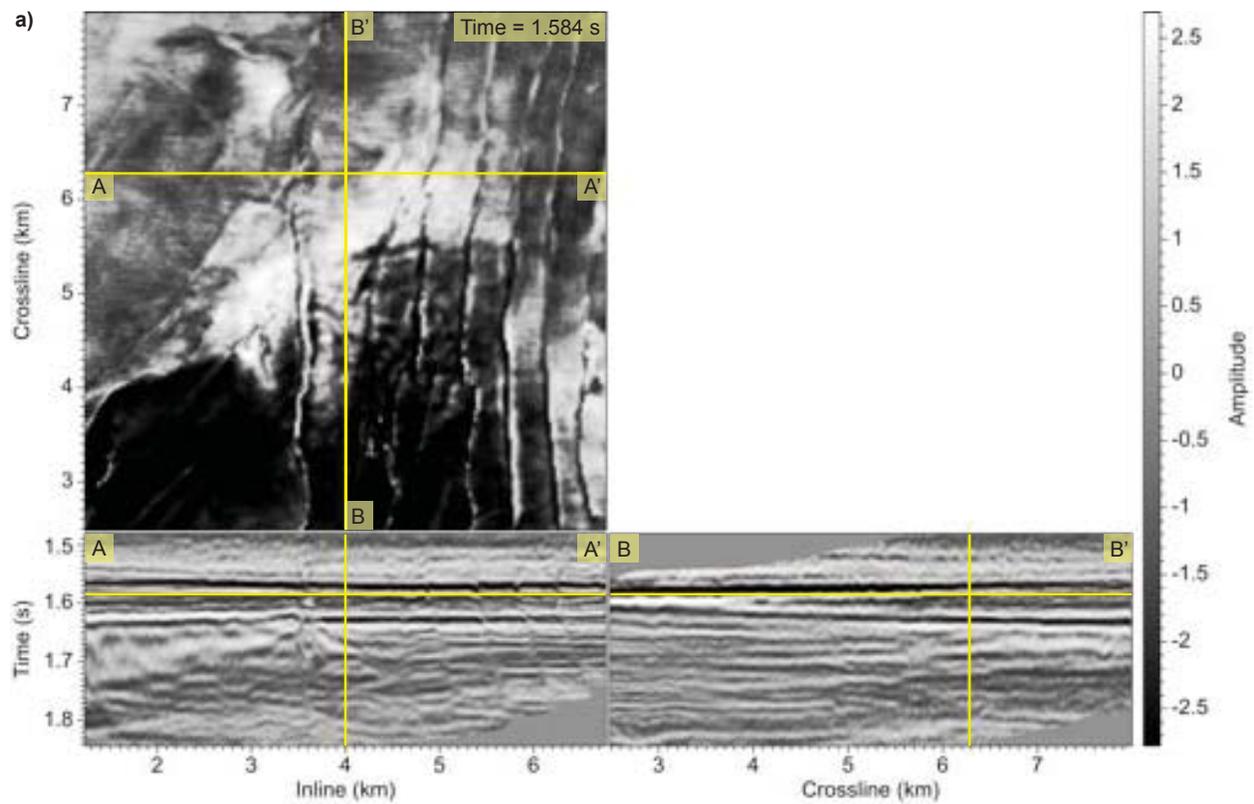


Figure 5.4: The seismic image shown in Figure 5.3(a) is unfaulted and unfolded (a) using the composite shift vectors (b).

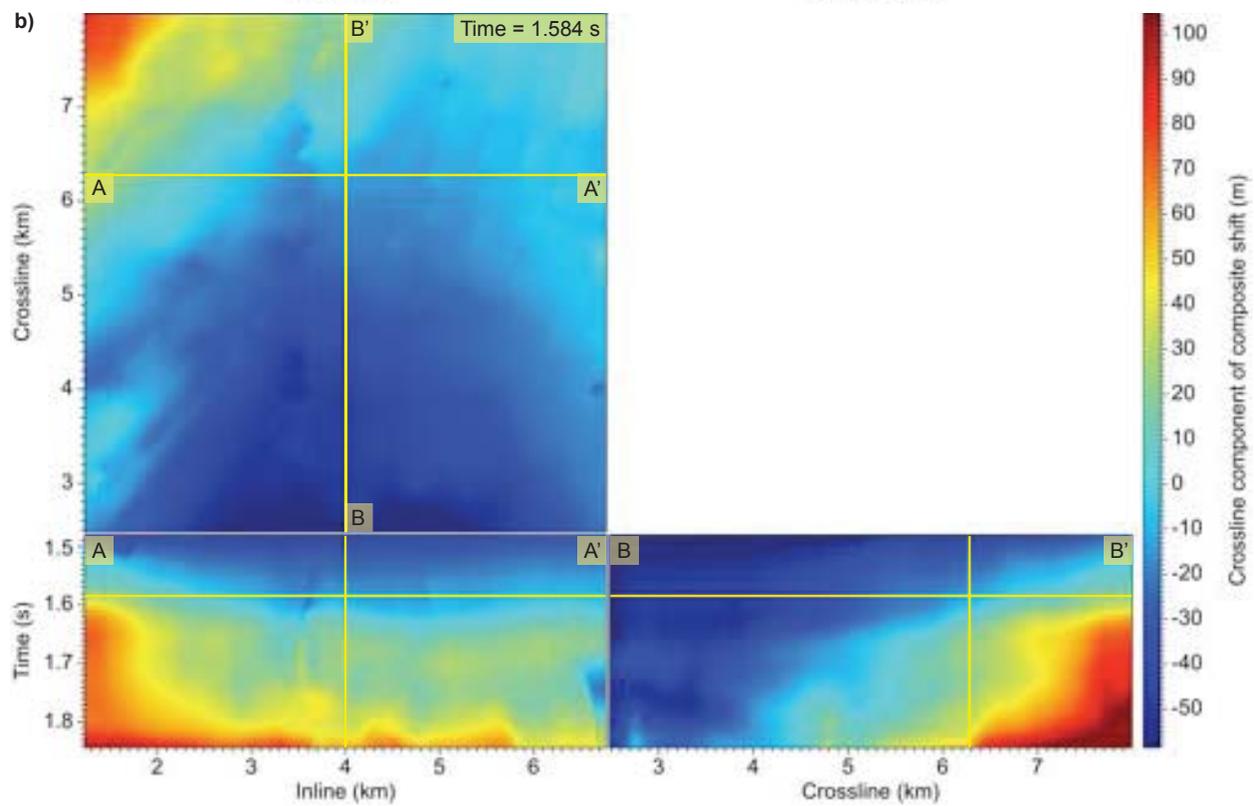
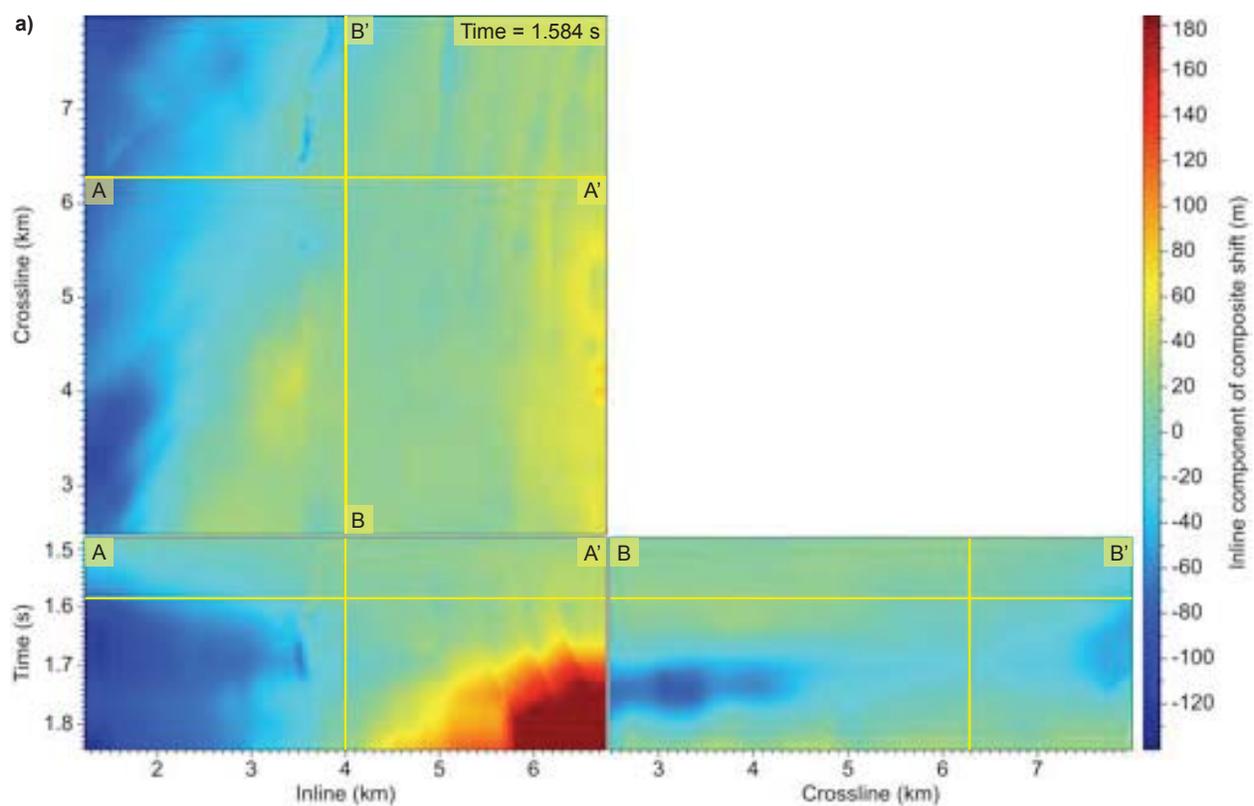


Figure 5.5: The inline (a) and crossline (b) components of the composite shift vectors used to unfault and unfold the seismic image shown in Figure 5.3(a).

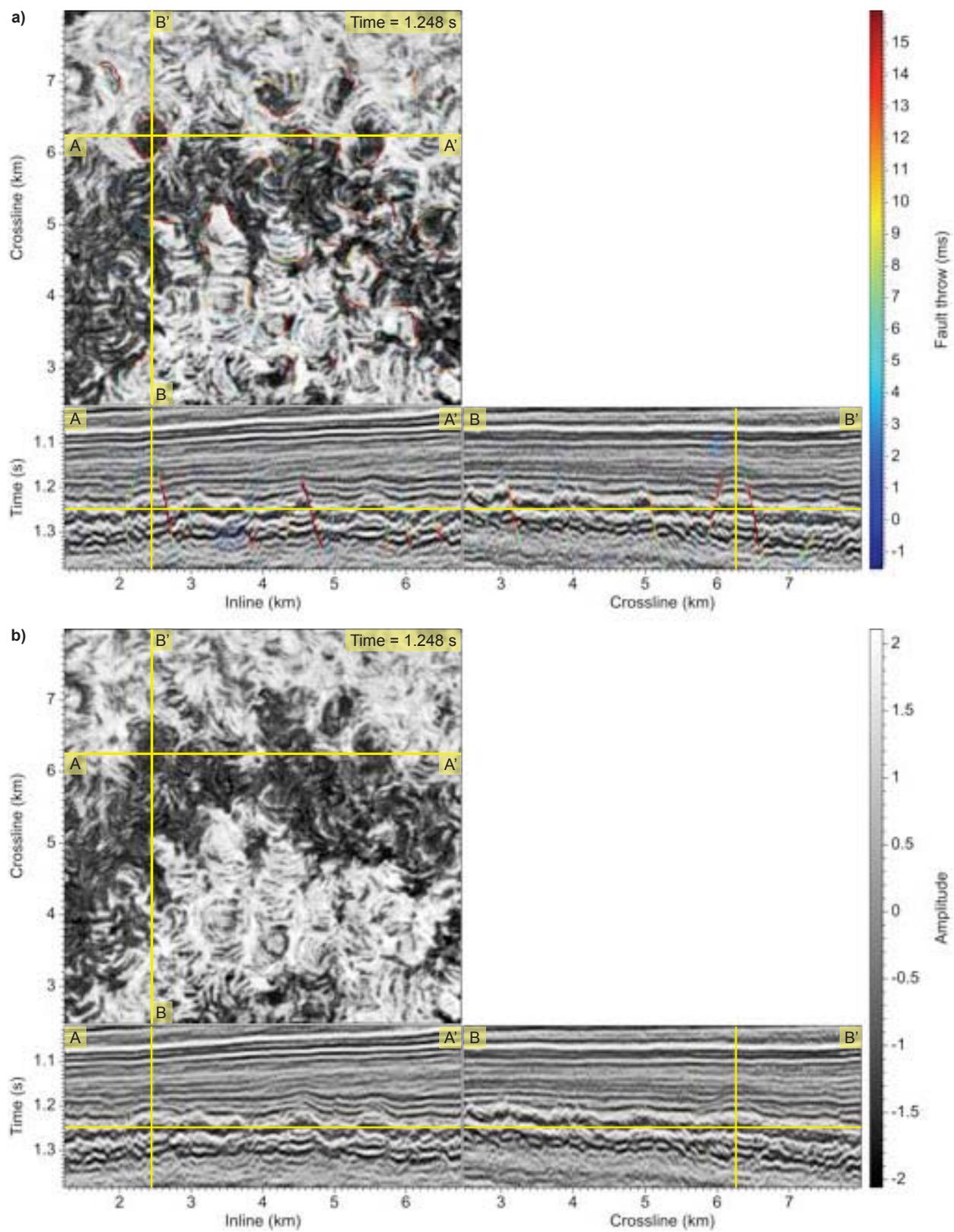


Figure 5.6: A seismic image (a) overlaid with the fault throw, and the unfaulted image (b).

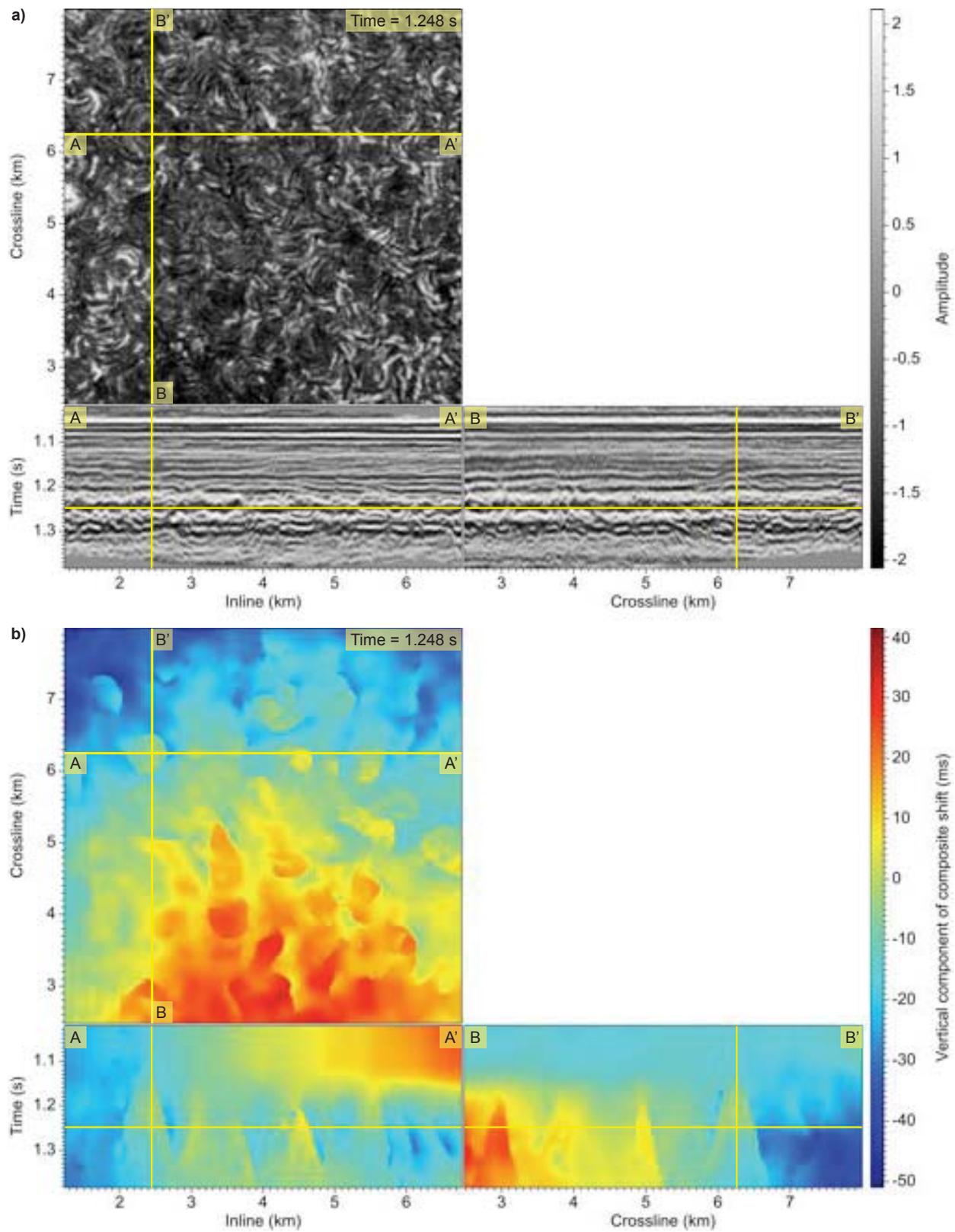


Figure 5.7: The seismic image shown in Figure 5.6(a) is unfaulted and unfolded (a) using the composite shift vectors (b).

## 5.4 Image unfolding

To unfold an image, we develop a new method for automatic image flattening (Lomask, 2003; Lomask *et al.*, 2006; Parks, 2010; Stark, 2004), which is used to compute seismic images in which sedimentary layering is horizontal. Most methods for automatic flattening are limited to only vertical shearing of an image. Flattening by vertical shear, however, can significantly distort image features (Luo & Hale, 2011). Moreover, for seismic images such as the one shown in Figure 5.2(a) with non-vertical faults, the true geologic deformation clearly is not vertical. For this reason, we do not limit our unfolding method to vertical shear only, but instead allow for non-vertical shift vectors.

To unfold an (unfaulted) image  $h(\mathbf{w})$ , we must find a mapping  $\mathbf{w}(\mathbf{u})$ , where  $\mathbf{u} = (u_1, u_2, u_3)$  are coordinates in the unfolded space, such that the image

$$g(\mathbf{u}) = h[\mathbf{w}(\mathbf{u})] \quad (5.2)$$

is flat, i.e., all horizons in the image are horizontal. We write the mapping  $\mathbf{w}(\mathbf{u})$  in terms of an unknown shift vector field  $\mathbf{r}(\mathbf{u})$ :

$$\mathbf{w}(\mathbf{u}) = \mathbf{u} - \mathbf{r}(\mathbf{u}), \quad (5.3)$$

which has a corresponding Jacobian matrix  $\mathbf{J} = \partial\mathbf{w}/\partial\mathbf{u}$ :

$$\mathbf{J} = \begin{bmatrix} 1 - \partial r_1 / \partial u_1 & -\partial r_1 / \partial u_2 & -\partial r_1 / \partial u_3 \\ -\partial r_2 / \partial u_1 & 1 - \partial r_2 / \partial u_2 & -\partial r_2 / \partial u_3 \\ -\partial r_3 / \partial u_1 & -\partial r_3 / \partial u_2 & 1 - \partial r_3 / \partial u_3 \end{bmatrix}. \quad (5.4)$$

The Jacobian matrix  $\mathbf{J}$  contains partial derivatives of the desired shift vector field  $\mathbf{r}(\mathbf{u})$ . Next, given unit normal vectors  $\mathbf{n} = [n_1 \ n_2 \ n_3]^\top$  that we compute for every image sample using structure tensors (Fehmers & Höcker, 2003; van Vliet & Verbeek, 1995), we can write the Jacobian matrix for rotation, which depends on the normal vectors:

$$\mathbf{J}_r = \begin{bmatrix} n_3 + n_2^2 / (1 + n_3) & -n_1 n_2 / (1 + n_3) & n_1 \\ -n_1 n_2 / (1 + n_3) & n_3 + n_1^2 / (1 + n_3) & n_2 \\ -n_1 & -n_2 & n_3 \end{bmatrix}. \quad (5.5)$$

The matrix  $\mathbf{J}_r$  is a rotation matrix for a rotation about an axis  $[n_2 \ -n_1 \ 0]^\top$  by an angle  $\arccos(n_3)$ . Normal vectors  $\mathbf{n}$  transform with the transpose of the Jacobian (Luo & Hale, 2011; Parks, 2010), so the transformed normal vectors  $\mathbf{m}$  (the normal vectors in the unfolded space) are given by

$$\mathbf{m} = \mathbf{J}^\top \mathbf{n}. \quad (5.6)$$

If all normal vectors  $\mathbf{m}$  point downward in the transformed image, i.e.,  $\mathbf{m} = [0 \ 0 \ 1]^\top$  everywhere, then the image is flat. It is straightforward to check that  $\mathbf{J}_r^\top \mathbf{n} = [0 \ 0 \ 1]^\top$ , so that if  $\mathbf{J} = \mathbf{J}_r$ , then applying the shifts  $\mathbf{r}(\mathbf{u})$  will unfold the image from which the normal vectors  $\mathbf{n}$  were computed.

Rather than solve  $\mathbf{J} = \mathbf{J}_r$  for the shifts  $\mathbf{r}(\mathbf{u})$  contained in  $\mathbf{J}$ , however, we instead choose to solve

$$\mathbf{J}^\top \mathbf{J}_r = \mathbf{I}, \quad (5.7)$$

because equation 5.7, as we will see, leads to explicit equations that determine the flatness of a transformed image  $g(\mathbf{u})$ . Solving equation 5.7 gives an approximately isometric mapping  $\mathbf{w}(\mathbf{u})$  that satisfies the so-called flexural slip constraints (Mallet, 2004), which assume that the deformation mechanism for geologic folding is flexural slip. Isometric mappings are desirable because they preserve metric properties. That is, if we could isometrically map an image to an unfolded flattened image, then all metric properties (e.g., length, angle, area, and volume) of features in the original image would be preserved in the unfolded image. Isometric mappings, however, exist only in special cases (Floater & Hormann, 2005), so in general, we solve for a mapping  $\mathbf{w}(\mathbf{u})$  that is only approximately isometric.

The columns of  $\mathbf{J}$  contain vectors  $D_1\mathbf{w}$ ,  $D_2\mathbf{w}$ , and  $D_3\mathbf{w}$ :

$$\mathbf{J} = [ D_1\mathbf{w} \ D_2\mathbf{w} \ D_3\mathbf{w} ], \quad (5.8)$$

where

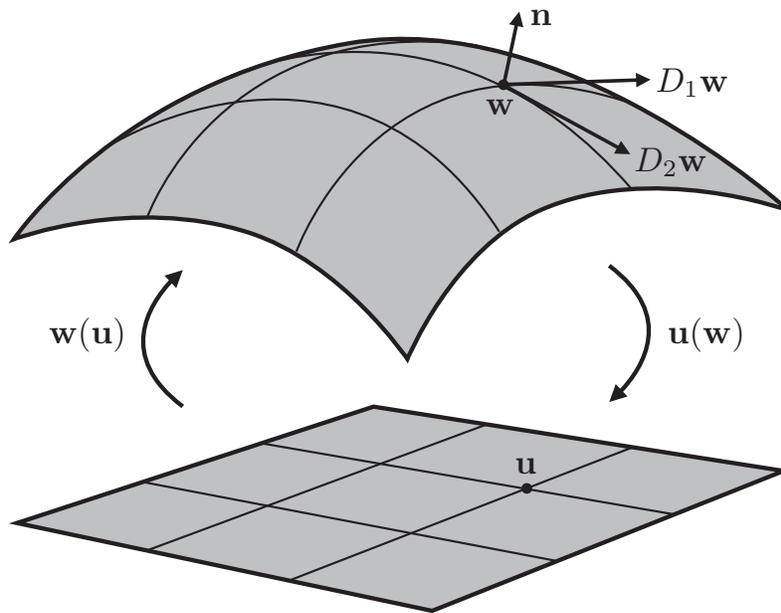


Figure 5.8: At any point  $\mathbf{w}$  on a smooth surface, there exist vectors  $D_1\mathbf{w}$  and  $D_2\mathbf{w}$  tangent to the surface and orthogonal to a normal vector  $\mathbf{n}$ . The mapping  $\mathbf{w}(\mathbf{u})$ , with inverse mapping  $\mathbf{u}(\mathbf{w})$ , determines local metric properties around points  $\mathbf{w}$  and  $\mathbf{u}$ . For seismic image flattening, we solve for an approximately isometric mapping  $\mathbf{w}(\mathbf{u})$  that flattens geologic horizons.

$$\begin{aligned}
D_1 \mathbf{w} &= \frac{\partial \mathbf{w}(\mathbf{u})}{\partial u_1} = \left( 1 - \frac{\partial r_1}{\partial u_1}, -\frac{\partial r_2}{\partial u_1}, -\frac{\partial r_3}{\partial u_1} \right), \\
D_2 \mathbf{w} &= \frac{\partial \mathbf{w}(\mathbf{u})}{\partial u_2} = \left( -\frac{\partial r_1}{\partial u_2}, 1 - \frac{\partial r_2}{\partial u_2}, -\frac{\partial r_3}{\partial u_2} \right), \\
D_3 \mathbf{w} &= \frac{\partial \mathbf{w}(\mathbf{u})}{\partial u_3} = \left( -\frac{\partial r_1}{\partial u_3}, -\frac{\partial r_2}{\partial u_3}, 1 - \frac{\partial r_3}{\partial u_3} \right).
\end{aligned} \tag{5.9}$$

Vectors  $D_1 \mathbf{w}$  and  $D_2 \mathbf{w}$  are tangent to a surface (e.g., a horizon) at  $\mathbf{u}$  and thus are orthogonal to a vector  $\mathbf{n}(\mathbf{u})$  normal to the surface at  $\mathbf{u}$  (see Figure 5.8). The vector  $D_3 \mathbf{w}$  is tangent to the line for which the horizontal coordinates  $u_1$  and  $u_2$  in the unfolded space are constant, i.e., the line in coordinates  $\mathbf{w}$  that maps to a vertical line in coordinates  $\mathbf{u}$  (Mallet, 2004). For an exactly isometric mapping  $\mathbf{w}(\mathbf{u})$ , tangent vectors  $D_1 \mathbf{w}$ ,  $D_2 \mathbf{w}$ , and  $D_3 \mathbf{w}$  are orthonormal vectors, and the corresponding Jacobian matrix is orthogonal. Next, if we denote the columns of  $\mathbf{J}_r$  by  $D_1 \hat{\mathbf{w}}$ ,  $D_2 \hat{\mathbf{w}}$ , and  $D_3 \hat{\mathbf{w}}$ , then

$$\mathbf{J}_r = [ D_1 \hat{\mathbf{w}} \quad D_2 \hat{\mathbf{w}} \quad D_3 \hat{\mathbf{w}} ], \tag{5.10}$$

and equation 5.7 states

$$\begin{bmatrix} (D_1 \mathbf{w})^\top (D_1 \hat{\mathbf{w}}) & (D_1 \mathbf{w})^\top (D_2 \hat{\mathbf{w}}) & (D_1 \mathbf{w})^\top (D_3 \hat{\mathbf{w}}) \\ (D_2 \mathbf{w})^\top (D_1 \hat{\mathbf{w}}) & (D_2 \mathbf{w})^\top (D_2 \hat{\mathbf{w}}) & (D_2 \mathbf{w})^\top (D_3 \hat{\mathbf{w}}) \\ (D_3 \mathbf{w})^\top (D_1 \hat{\mathbf{w}}) & (D_3 \mathbf{w})^\top (D_2 \hat{\mathbf{w}}) & (D_3 \mathbf{w})^\top (D_3 \hat{\mathbf{w}}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.11}$$

Note that vectors  $D_1 \hat{\mathbf{w}}$ ,  $D_2 \hat{\mathbf{w}}$  and  $D_3 \hat{\mathbf{w}}$  are orthonormal, and so  $\mathbf{J}_r$  is an orthogonal matrix. The matrix  $\mathbf{J}^\top \mathbf{J}_r$  characterizes local metric properties such as length, angle, area, and volume, so by setting this matrix equal to the identity, we constrain the type of deformation parameterized by the mapping  $\mathbf{w}(\mathbf{u})$ . Equation 5.11 gives nine linear equations for three unknowns—the components of the shift vector field  $\mathbf{r}(\mathbf{u})$ —to be solved at each image sample:

$$\begin{aligned}
n_1 \left( 1 - \frac{\partial r_1}{\partial u_1} \right) - n_2 \frac{\partial r_2}{\partial u_1} - n_3 \frac{\partial r_3}{\partial u_1} &= 0, \\
-n_1 \frac{\partial r_1}{\partial u_2} + n_2 \left( 1 - \frac{\partial r_2}{\partial u_2} \right) - n_3 \frac{\partial r_3}{\partial u_2} &= 0,
\end{aligned} \tag{5.12}$$

and

$$\begin{aligned}
\alpha \left(1 - \frac{\partial r_1}{\partial u_1}\right) - \gamma \frac{\partial r_2}{\partial u_1} + n_1 \frac{\partial r_3}{\partial u_1} &= 1, \\
\gamma \left(1 - \frac{\partial r_1}{\partial u_1}\right) - \beta \frac{\partial r_2}{\partial u_1} + n_2 \frac{\partial r_3}{\partial u_1} &= 0, \\
-\alpha \frac{\partial r_1}{\partial u_2} + \gamma \left(1 - \frac{\partial r_2}{\partial u_2}\right) + n_1 \frac{\partial r_3}{\partial u_2} &= 0, \\
-\gamma \frac{\partial r_1}{\partial u_2} + \beta \left(1 - \frac{\partial r_2}{\partial u_2}\right) + n_2 \frac{\partial r_3}{\partial u_2} &= 1,
\end{aligned} \tag{5.13}$$

and

$$\begin{aligned}
-\alpha \frac{\partial r_1}{\partial u_3} - \gamma \frac{\partial r_2}{\partial u_3} - n_1 \left(1 - \frac{\partial r_3}{\partial u_3}\right) &= 0, \\
-\gamma \frac{\partial r_1}{\partial u_3} - \beta \frac{\partial r_2}{\partial u_3} - n_2 \left(1 - \frac{\partial r_3}{\partial u_3}\right) &= 0, \\
-n_1 \frac{\partial r_1}{\partial u_3} - n_2 \frac{\partial r_2}{\partial u_3} + n_3 \left(1 - \frac{\partial r_3}{\partial u_3}\right) &= 1,
\end{aligned} \tag{5.14}$$

where

$$\begin{aligned}
\alpha &= n_3 + n_2^2/(1 + n_3), \\
\beta &= n_3 + n_1^2/(1 + n_3), \\
\gamma &= -n_1 n_2/(1 + n_3).
\end{aligned} \tag{5.15}$$

In total, we have  $9N$  equations for  $3N$  unknowns, where  $N$  is the number of image samples.

These equations can be represented in matrix form as

$$\mathbf{WAGr} = \mathbf{b}, \tag{5.16}$$

where  $\mathbf{r}$  is a  $(3N \times 1)$  unknown shift vector,  $\mathbf{G}$  is a  $(9N \times 3N)$  matrix containing finite-difference derivative approximations,  $\mathbf{A}$  is a  $(9N \times 9N)$  diagonal matrix containing coefficients specified in equations 5.12 through 5.14,  $\mathbf{W}$  is a  $(9N \times 9N)$  diagonal matrix containing weights for each equation, and  $\mathbf{b}$  is a  $(9N \times 1)$  vector containing known constants specified in equations 5.12 through 5.14. Then, the least-squares solution of equation 5.16 satisfies

$$(\mathbf{WAG})^\top \mathbf{WAGr} = (\mathbf{WAG})^\top \mathbf{b}. \tag{5.17}$$

The matrices  $\mathbf{W}$ ,  $\mathbf{A}$ , and  $\mathbf{G}$  are sparse, and in practice we do not explicitly form these matrices. We solve equation 5.17 using a conjugate gradient solver, which requires only the computation of vectors  $\mathbf{WAGy}$  and  $(\mathbf{WAG})^\top \mathbf{z}$ , where  $\mathbf{y}$  and  $\mathbf{z}$  are arbitrary  $(3N \times 1)$  and  $(9N \times 1)$  vectors, respectively.

Equation 5.7 describes an isometric mapping of an image to a unfolded image, but in general, we cannot expect to find an exactly isometric mapping for all images. In fact, the only image for which we can find an exactly isometric mapping is one in which the normal vectors are constant. For all other images, equations 5.12, 5.13, and 5.14 cannot be satisfied exactly, and we must decide which equations to emphasize.

Equations 5.12, 5.13, and 5.14 correspond to entries in the matrix  $\mathbf{J}^\top \mathbf{J}_r$  on the left side of equation 5.11, and they characterize the lengths of and angles between tangent vectors  $D_1 \mathbf{w}$ ,  $D_2 \mathbf{w}$ , and  $D_3 \mathbf{w}$ . Specifically, the diagonal entries of the metric tensor characterize lengths of tangent vectors, while the off-diagonal entries characterize angles between tangent vectors. For image unfolding, we give most weight to equations 5.12, which determine the angle between the surface tangent vectors  $D_1 \mathbf{w}$  and  $D_2 \mathbf{w}$  and the normal vector  $\mathbf{n}(\mathbf{u})$ . If these equations are satisfied, then the image  $g(\mathbf{u}) = h(\mathbf{w}(\mathbf{u})) = h(\mathbf{u} - \mathbf{r}(\mathbf{u}))$  obtained by applying the shifts  $\mathbf{r}(\mathbf{u})$  will be flat, and all horizons will be horizontal.

We give less weight to equations 5.13. The four corresponding entries in the matrix  $\mathbf{J}^\top \mathbf{J}_r$  resemble what is referred to as the *first fundamental form* (Floater & Hormann, 2005), which characterizes lengths, areas, and angles measured on a surface. If the first fundamental form equals the identity, then the surface is said to be locally *developable*, meaning it is isometric to a plane. Assuming that a geologic horizon was initially deposited as a horizontal layer (i.e., a plane), a developable geologic horizon would indicate that metric properties measured on the horizon have been preserved since its initial deposition.

Finally, we give least weight to equations 5.14, which determine the length of tangent vector  $D_3 \mathbf{w}$  and the angles it forms with surface tangent vectors. Recall that  $D_3 \mathbf{w}$  is tangent to the line in coordinates  $\mathbf{w}$  that maps to a vertical line in the unfolded coordinates  $\mathbf{u}$ . Thus,

if  $D_3\mathbf{w}$  is a unit vector parallel to the normal vector  $\mathbf{n}$ , then the thickness of sedimentary layers measured perpendicular to bedding will be preserved in the unfolding process. For most images, we cannot preserve thickness while unfolding, so we give the corresponding equations least weight.

Figure 5.4(a) and Figure 5.7(a) show unfolded unfaulted images computed from the unfaulted images shown in Figure 5.3(b) and Figure 5.6(b), respectively. The weights given to equations 5.12, 5.13, and 5.14 for both examples were 1.0, 0.1, and 0.01, respectively. Normal vectors were computed using structure tensors with Gaussian smoothing filters (Hale, 2009c) with a vertical half-width of 32 ms and horizontal half-widths of 50 m. An additional weight computed from eigenvalues of the structure tensor provides a measure of the planarity of image features, and is used to give more weight to coherent, locally planar image features. In addition, when solving equations 5.12, 5.13, and 5.14, we used smoothing preconditioners as described by Parks (2010) to speed convergence. The smoothing filter in the vertical direction had a half-width of 24 ms, while the filters in the horizontal directions had half-widths of 150 m.

## 5.5 Horizon extraction

Once we obtain an unfolded unfaulted image, the corresponding unfolding shift vectors, and the interpolated dip-separation vectors, we can extract geologic horizons such as those shown in Figure 5.9.

We extract a horizon by first selecting a horizontal slice of constant  $u_3$  in an unfolded unfaulted image  $g(\mathbf{u})$ . Next we form a composite mapping  $\mathbf{x}(\mathbf{u})$  by combining the mapping  $\mathbf{x}(\mathbf{w}) = \mathbf{w} + \mathbf{t}(\mathbf{w})$  used to unfault an image with the mapping  $\mathbf{w}(\mathbf{u}) = \mathbf{u} - \mathbf{r}(\mathbf{u})$  used to unfold an image to obtain

$$\mathbf{x}(\mathbf{u}) = \mathbf{u} - \mathbf{s}(\mathbf{u}), \quad (5.18)$$

where  $\mathbf{s}(\mathbf{u})$  are the composite shift vectors

$$\mathbf{s}(\mathbf{u}) = \mathbf{r}(\mathbf{u}) - \mathbf{t}(\mathbf{u} - \mathbf{r}(\mathbf{u})), \quad (5.19)$$

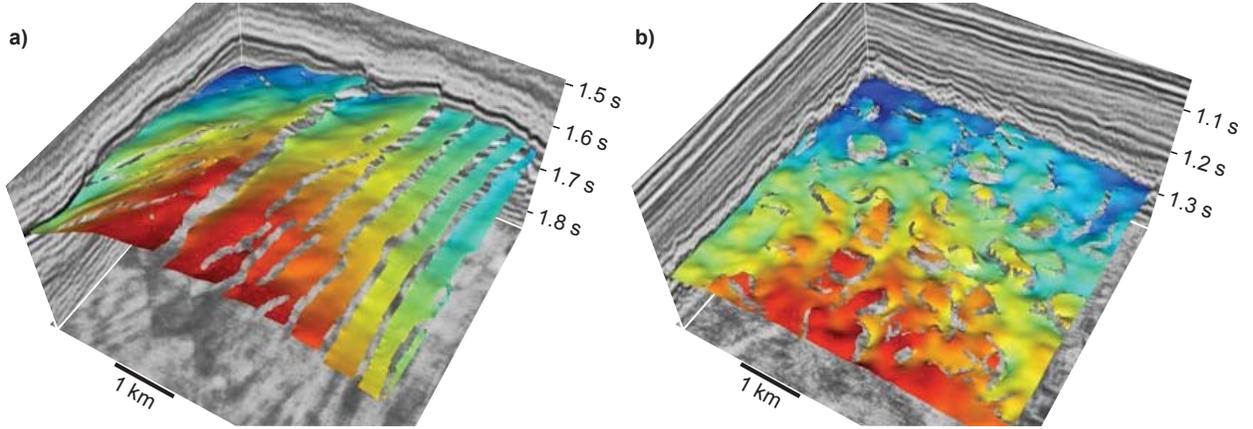


Figure 5.9: Geologic horizons extracted using the composite shift vector fields shown in Figure 5.4(b) and Figure 5.5(a) and Figure 5.7(b) (b). On each horizon, colors represent the vertical component of the corresponding composite shift vector field.

which allows for a direct mapping from an image  $f(\mathbf{x})$  to an unfolded unfaulted image  $g(\mathbf{u})$  with

$$g(\mathbf{u}) = f(\mathbf{u} - \mathbf{s}(\mathbf{u})). \quad (5.20)$$

The image  $g(\mathbf{u})$  is computed from  $f(\mathbf{x})$  using 3D sinc interpolation. We used equation 5.20 to compute the unfolded unfaulted image shown in Figure 5.4(a) from the seismic image shown in Figure 5.3(a) using the composite shift vectors whose vertical component is shown in Figure 5.4(b) and whose inline and crossline components are shown in Figure 5.5(a) and Figure 5.5(b), respectively. Similarly, Figure 5.7(a) shows the unfolded unfaulted image computed from the seismic image shown in Figure 5.6(a) using the composite shift vectors shown in Figure 5.6(b). Note that in the chaotic region located between 1.2 and 1.4 seconds in Figure 5.6(a), it is difficult to estimate accurate normal vectors, and for this reason some of the small-scale folds in this region are not completely unfolded. Using the composite shift vectors  $\mathbf{s}(\mathbf{u})$ , we map a surface of constant  $u_3$ , which corresponds to constant geologic time or constant depositional time, to a geologic horizon in present-day coordinates. Figure 5.9(a) shows a geologic horizon extracted from the composite shift vectors shown in Figure 5.4(b) and Figure 5.5, while Figure 5.9(b) shows a geologic horizon extracted from the composite

shift vectors shown in Figure 5.7(b). In the horizon in Figure 5.9(a), notice the en échelon faults that can also be clearly seen in the seismic image in Figure 5.3(a). In the horizon in Figure 5.9(b), notice the roughly circular fault polygons, which correspond to the conical fault surfaces described by Hale (2013b). Note that although we show only a single horizon for each image, it is trivial to extract another horizon by simply choosing a different horizontal slice of constant  $u_3$  from an unfolded unfaulted image.

## 5.6 Conclusion

We have presented a method to automatically unfault and unfold seismic images. The method requires an estimate of fault locations and dip-separation vectors, which, for the examples shown in this paper, we obtain using the method described by Hale (2013b). Unfolded, unfaulted images are images in which horizontal slices correspond to a constant geologic time, and the geologic age of horizontal slices increases with increasing time. The cost of the method depends somewhat on the complexity of the geology in an image, but for the previous examples which contained  $90 \times 221 \times 220$  image samples, the total runtime was on the order of a few minutes on an 8-core workstation, which was roughly the same amount of time required to locate faults and estimate dip-separation vectors using Hale’s (2012) method. For an image with  $1000 \times 1000 \times 1000$  samples, the total runtime is on the order of a few hours.

Some limitations remain in our method. One is that we are currently using only the estimated dip separation to unfault an image, but in reality, fault slip consists of both dip and strike separation. Strike separation, however, is more difficult to estimate not only because strike separation tends to be parallel to sedimentary layering, but also because seismic images have lower lateral than vertical resolution. There are also difficulties with estimating dip separation for certain fault geometries, e.g., intersecting faults, which require multiple dip-separation vectors at a single location on a fault surface to accurately describe the faulting process. Another limitation arises from the way in which we compute normal vectors. Because normal vectors are computed in local windows (and, moreover, are constrained to

point in the positive time or depth direction), they cannot distinguish overturned or recumbent horizons. This limitation could be overcome by using a different method for computing normal vectors or by filtering normal vectors computed using structure tensors to distinguish overturned or recumbent horizons.

Our two-step process of image unfauling followed by image unfolding might suggest that true geologic deformation is also a two-step process, i.e., deposited sediments are first deformed during a period of folding that is temporally separate from a subsequent period of faulting. Of course this is not the case, as faulting and folding do not occur as distinct events. Thus, our method does not truly reverse the geologic deformation resulting from, for example, syntectonic sedimentation, although it is still able to unfaul, unfold, and extract horizons from images containing such deformation.

Although our method consists of two steps—image unfauling followed by image unfolding—it is possible to combine these two steps into one, by solving for shifts that unfold an image while constraining the shift vectors to be equal to dip-separation vectors estimated at fault locations, or alternatively by changing the equations we solve at the known locations of fault surfaces. Recall that when unfolding an image, we try to preserve surface metric properties (equation 5.13) and sedimentary layer thickness (equation 5.14), while emphasizing image flatness (equation 5.12). The unfolding process, however, can only preserve surface metric properties and layer thickness with respect to the input image, which, for the examples shown in this paper, was an unfaulked image. Preserving surface metric properties and layer thickness with respect to the original faulted and folded image would be more appropriate, but to do so would require a one-step image unfauling and unfolding process.

## 5.7 Acknowledgments

Thanks to dGB Earth Sciences for providing the 3D seismic images shown in this paper. Thanks also to Kurt Marfurt, Shankar Mitra, and an anonymous reviewer for helpful comments and suggestions. This research was supported by the sponsors of the Center for Wave Phenomena at the Colorado School of Mines.

## CHAPTER 6

### SUMMARY AND SUGGESTIONS

Seismic images provide relevant geologic information that aid in our understanding of the earth's subsurface. In this thesis, I developed improved methods for reflection seismic data processing, including methods for velocity estimation, seismic migration, and interpretation of seismic images. The main results of this thesis, as well as suggestions for future research, are summarized below.

Performing velocity analysis to estimate the velocity model is a key step in processing reflection seismic data. In Chapter 2, I proposed a method for NMO velocity analysis using weighted semblance spectra. The weighting function introduced in the semblance calculation was designed to minimize semblance, and, as was demonstrated, this weighting function increased not only the resolution but also the accuracy of the resulting semblance spectra. The weighting function used in the semblance calculation was derived by analyzing the change in NMO time for a small change in velocity, i.e., the weight was a function of both offset and time. For future work, it would be worthwhile to investigate other weighting functions, for example, those with weights that depend on reflection angle rather than offset, that might offer superior sensitivity to changes in velocity.

Weighted semblance can also be viewed as a weighted stack. With this in mind, we could follow a similar approach with other types of gathers besides common midpoint gathers. For example, by introducing a weighting function when stacking common image or common angle gathers, we could potentially increase the resolution of the resulting seismic image. This would improve our ability to identify and interpret geologic features from seismic images, and could also serve to reduce the uncertainty in information obtained from images.

In Chapter 3, I demonstrated that, by computing a seismic image by minimizing the difference between predicted data and time-shifted observed data, I was able to use an

erroneous velocity model to obtain an image with features similar to those computed using a more accurate velocity. A primary motivation for this work was the observation that errors in the velocity models we compute, which are unavoidable in practice, lead to differences between the traveltimes of predicted and observed data that, in turn, degrade the seismic image. However, it is worth noting that traveltime differences generally result from more than just velocity errors. For example, an inaccurate assumption of the model of wave propagation in the subsurface can also lead to significant traveltime differences. In fact, one of the primary motivations for using a more accurate, and typically more expensive, model of wave propagation is the resulting reduction in traveltime differences and the corresponding improvement in the quality and accuracy of migration images.

The examples shown in Chapter 3 were computed by solving the acoustic constant-density wave equation in two dimensions. The earth, of course, is neither acoustic with constant density, nor two-dimensional. Thus, areas for future research include the extension of the method proposed in Chapter 3 to more accurate wave equations, and the extension to three dimensions. Although both extensions would require an increase in computational effort, I expect them to otherwise be straightforward.

Another potential topic for future research is the extension of the method for migration proposed in Chapter 3 to velocity analysis. Following the computation of an image using our proposed method, there remained significant differences in traveltime between predicted and observed seismic data. These traveltime differences were indicative of the errors in the velocity model used for migration. Thus, to update the velocity model, we might invert the traveltimes to find the optimal velocity model for which the traveltimes of predicted and observed data best match.

In Chapters 4 and 5, I proposed methods for automatic horizon extraction to aid in the interpretation of seismic images. In Chapter 4, I presented a novel method for seismic image flattening using non-vertical shifts, which are necessary to flatten without distorting image features resulting from non-vertical geologic deformation. An advantage of the proposed

method is its flexibility; one has the ability to flatten an image by vertical shear, by rotation, or by a combination of vertical shear and rotation of image features. Unfortunately, this flexibility also presents a limitation, because the true type of deformation must be known and specified beforehand in order for an image to be flattened according to the true geologic deformation. In practice, it might be difficult to determine the true deformation.

However, the framework for seismic image flattening proposed in Chapter 4 could perhaps be used to invert for the type of geologic deformation that occurred. One approach might be to seek the deformation that minimizes the distortion of certain image features, for example, to solve for the deformation that best preserves image features within geologic horizons. Other approaches could also be devised, given some a priori knowledge of the type of geologic deformation that occurred.

Due in part to the difficulty in determining the true geologic deformation, in Chapter 5, I proposed a method for unfauling and unfolding seismic images, in which the unfolding step was performed by solving for unfolding vector shifts that yield an approximately isometric mapping between a seismic image and the corresponding flattened image. Choosing to always solve for an approximately isometric mapping eliminates the need to specify the type of deformation, while also minimizing the distortion of image features while flattening.

For the method proposed in Chapter 5, a seismic image is first unfaulked by shifting image features according to a vector field computed by estimating (at fault locations) and interpolating (between fault locations) dip-separation vectors. One limitation of this approach is that all faults within a seismic image are unfaulked simultaneously. In reality, the faulting observed in an image may have occurred in distinct episodes at different times. As a result, some fault systems created by multiple episodes of faulting cannot be correctly unfaulked with the proposed approach. For example, intersecting faults cannot be correctly unfaulked by simultaneously shifting image features according to interpolated dip-separation vectors.

After unfaulking, an image is unfolded by solving for an approximately isometric mapping that best preserves metric properties such as angle, length, and volume, while also flattening

the image. Seeking to preserve metric properties while flattening minimizes the distortion of image features. With our approach, however, these properties are preserved with respect to only the input seismic image, which was an unfaulted image in the examples shown in Chapter 5. This presents another limitation, because ideally we should seek to preserve metric properties, and thereby minimize the distortion of image features, with respect to the present-day seismic image prior to unfaulting.

One way to address the two limitations mentioned above is to unfault and unfold a seismic image in one single step, perhaps by constraining the unfolding shift vectors to be equal to estimated dip-separation vectors at fault locations, or by constraining corresponding image samples separated by a fault to lie on the same geologic horizon. In any case, it is more appropriate to minimize distortions of image features relative to the seismic image of the present-day subsurface, rather than an unfaulted image.

## REFERENCES CITED

- Admasu, F. 2008. *A stochastic method for automated matching of horizons across a fault in 3D seismic data*. Ph.D. thesis, Otto-von-Guericke-University Magdeburg.
- Al-Yahya, K. 1989. Velocity analysis by iterative profile migration. *Geophysics*, **54**, 718–729.
- Aurnhammer, M., & Tönnies, K. D. 2005. A genetic algorithm for automated horizon correlation across faults in seismic images. *IEEE Transactions on Evolutionary Computation*, **9**(2), 201–210.
- Baysal, E., Kosloff, D., & Sherwood, J. 1983. Reverse time migration. *Geophysics*, **48**(11), 1514–1524.
- Bednar, J. B., Shin, C., & Pyun, S. 2007. Comparison of waveform inversion, part 2: phase approach. *Geophys. Prospect.*, **55**, 455–465.
- Biondi, B., & Sava, P. 1999. Wave-equation migration velocity analysis. *69th Annual International Meeting, SEG, Expanded Abstracts*, 1723–1726.
- Borgos, H. G., Skov, T., Randen, T., & Sonneland, L. 2003. Automated geometry extraction from 3D seismic data. *73rd Annual International Meeting, SEG, Expanded Abstracts*.
- Celis, V., & Larner, K. 2002. *Selective-correlation velocity analysis*. M.Sc. thesis, CWP-434. Center for Wave Phenomena, Colorado School of Mines.
- Chavent, G., & Plessix, R. 1999. An optimal true-amplitude least-squares prestack depth-migration operator. *Geophysics*, **64**(2), 508–515.
- Choi, Y., & Alkhalifah, T. 2011. Frequency-domain Waveform Inversion Using the Unwrapped Phase. *81st Annual International Meeting, SEG, Expanded Abstracts*.
- Claerbout, J. F. 1992. *Earth Soundings Analysis: Processing versus Inversion*. Blackwell Scientific Publications.
- Cohen, J. K., & Bleistein, N. 1979. Velocity inversion procedure for acoustic waves. *Geophysics*, **44**(6), 1077–1087.
- Corcoran, C. T., & Seriff, A. J. 1993. Methods for processing seismic data. *U.S. Patent 5,197,039; European Patent 335-450*.

- Dai, W. 2012. *Multisource Least-squares Reverse Time Migration*. Ph.D. thesis, King Abdullah University of Science and Technology.
- Fehmers, G. C., & Höcker, C. F. W. 2003. Fast structural interpretations with structure-oriented filtering. *Geophysics*, **68**, 1286–1293.
- Floater, M. S., & Hormann, K. 2005. Surface Parameterization: A Tutorial and Survey. *Advances in Multiresolution for Geometric Modelling*, 157–186.
- Fomel, S. 2009. Velocity analysis using AB semblance. *Geophysical Prospecting*, **57**, 311–321.
- Fowler, P. 1985. *Migration velocity analysis by optimization*. Tech. rept. SEP-44. Stanford Exploration Project, Stanford University.
- Gray, S. 1997. True amplitude migration: a comparison of three approaches. *Geophysics*, **62**, 929–936.
- Guitten, A. 2004. Amplitude and kinematic corrections of migrated images for nonunitary imaging operators. *Geophysics*, **69**(4), 1017–1024.
- Hale, D. 2009a. Image-guided blended neighbor interpolation. *CWP Report 634*.
- Hale, D. 2009b. *Structure-oriented smoothing and semblance*. Tech. rept. CWP-635. Center for Wave Phenomena, Colorado School of Mines.
- Hale, D. 2009c. Structure-oriented smoothing and semblance. *CWP Report 635*.
- Hale, D. 2013a. Dynamic warping of seismic images. *Geophysics*, **78**(2), S105–S115.
- Hale, D. 2013b. Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3D seismic images. *Geophysics*, **78**(2), O33–O43.
- Hicks, G., & Pratt, R. G. 2001. Reflection waveform inversion using local descent methods: Estimating attenuation and velocity over a gassand deposit. *Geophysics*, **66**(2), 598–612.
- Hill, S. J., & Rüger, A. 2008. *Illustrated Seismic Processing*.
- Howard, R. E. 1990. *Method for attribute tracking in seismic data: US Patent 5,056,066*.
- Kamei, R., Brenders, A. J., & Pratt, R. G. 2011. A discussion on the advantages of phase-only waveform inversion in the Laplace-Fourier domain: Validation with marine and land seismic data. *81st Annual International Meeting, SEG, Expanded Abstracts*.

- Kühl, H., & Sacchi, M. D. 2003. Least-squares wave-equation migration for AVP/AVA inversion. *Geophysics*, **68**, 262–273.
- Lailly, P. 1983. The seismic inverse problem as a sequence of before stack migration. *Conference on Inverse Scattering: Theory and Applications*, SIAM, 206–220.
- Lailly, P., & Versteeg, R. 1990. The Marmousi workshop – Introduction. *Proceedings of the 52nd EAEG Workshop – Practical Aspects of Seismic Data Inversion*.
- Levin, S. 1984. Principle of reverse-time migration. *Geophysics*, **49**(5), 581–583.
- Liang, L., Hale, D., & Maučec, M. 2010. Estimating fault displacements in seismic images. *80th Annual International Meeting, SEG, Expanded Abstracts*, 1357–1361.
- Loewenthal, D., & Mufti, I. 1983. Reversed time migration in spatial frequency domain. *Geophysics*, **48**(5), 627–635.
- Lomask, J. 2003. Flattening 3-D seismic cubes without picking. *73rd Annual International Meeting, SEG, Expanded Abstracts*.
- Lomask, J., Guitton, A., Fomel, S., Claerbout, J. F., & Valenciano, A. 2006. Flattening without picking. *Geophysics*, **71**, 13–20.
- Luo, S., & Hale, D. 2011. Non-vertical deformations for seismic image flattening. *CWP Report 694*.
- Ma, Y. 2012. *Waveform-based velocity estimation from reflection seismic data*. Ph.D. thesis, Colorado School of Mines.
- Mallet, J.-L. 2004. Space-Time Mathematical Framework for Sedimentary Geology. *Mathematical Geology*, **36**, 1–32.
- McMechan, G. A. 1983. Migration by extrapolation of time-dependent boundary values. *Geophys. Prospect.*, **31**(3), 413–420.
- Mulder, W. A., & Plessix, R.-E. 2004. A comparison between one-way and two-way wave-equation migration. *Geophysics*, **69**(6), 1491–1504.
- Neidell, N. S., & Taner, M. T. 1971. Semblance and other coherency measures for multi-channel data. *Geophysics*, **36**, 482–297.
- Nemeth, T., Wu, C., & Schuster, G. T. 1999. Least-squares migration of incomplete reflection data. *Geophysics*, **64**, 208–221.

- Nocedal, J., & Wright, S. J. 2000. *Numerical Optimization*. Springer Science+Business Media, LLC.
- Østmo, S., & Plessix, R.-E. 2002. Finite-difference iterative migration by linearized waveform inversion in the frequency domain. *72nd Annual International Meeting, SEG, Expanded Abstracts*.
- Park, S. W., Linsen, L., Kreylos, O., Owens, J. D., & Hamann, B. 2006. Discrete Sibson interpolation. *IEEE Transactions on Visualization and Computer Graphics*, **12**(2), 243–253.
- Parks, D. 2010. *Seismic image flattening as a linear inverse problem*. M.Sc. thesis CWP-643. Center for Wave Phenomena, Colorado School of Mines.
- Plessix, R.-E., & Mulder, W. A. 2004. Frequency-domain finite-difference amplitude-preserving migration. *Geophys. J. Int.*, **157**, 975–987.
- Pratt, R. G., Shin, C., & Hicks, G. J. 1998. Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion. *Geophys. J. Int.*, **113**, 341–462.
- Rickett, J. E. 2003. Illumination-based normalization for wave-equation depth migration. *Geophysics*, **68**, 1371–1379.
- Sakoe, H., & Chiba, S. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **26**, 43–49.
- Sarkar, D., Castagna, J. P., & Lamb, W. 2000. AVO and velocity analysis. *Geophysics*, **66**, 1284–1293.
- Sarkar, D., Baumel, R. T., & Larner, K. L. 2002. Velocity analysis in the presence of amplitude variation. *Geophysics*, **67**, 1664–1672.
- Sava, P., & Biondi, B. 2004. Wave-equation migration velocity analysis. I: Theory. *Geophys. Prospect.*, **52**, 593–606.
- Shen, P. 2004. *Wave equation migration velocity analysis by differential semblance optimization*. Ph.D. thesis, Rice University.
- Shin, C., & Min, D. J. 2006. Waveform inversion using a logarithmic wavefield. *Geophysics*, **71**, R31–R42.
- Shin, C., Jang, S., & Min, D. J. 2001. Improved amplitude preservation for prestack depth migration by inverse scattering theory. *Geophys. Prospect.*, **49**, 592–606.

- Sibson, R. 1981. A brief description of natural neighbor interpolation. *Pages 223–233 of:* Marks, J. (ed), *Interpreting Multivariate Data*. John Wiley & Sons.
- Snieder, R., Xie, M., Pica, A., & Tarantola, A. 1989. Retrieving both the impedance contrast and background velocity: A global strategy for the seismic reflection problem. *Geophysics*, **54**(8), 991–1000.
- Stark, T. J. 1996. Surface slice generation and interpretation: A review. *The Leading Edge*, **15**(7), 818–819.
- Stark, T. J. 2004. Relative geologic time (age) volumes — Relating every seismic sample to a geologically reasonable horizon. *The Leading Edge*, **23**(9), 928–932.
- Symes, W. W. 2008. Approximate linearized inversion by optimal scaling of prestack depth migration. *Geophysics*, **73**, R23–R35.
- Symes, W. W. 2009. The seismic reflection inverse problem. *Inverse Problems*, **25**, 1–39.
- Symes, W. W., & Carazzone, J. 1991. Velocity inversion by differential semblance optimization. *Geophysics*, **56**(5), 654–663.
- Taner, M. T., & Koehler, F. 1969. Velocity spectra – digital computer derivation and applications of velocity functions. *Geophysics*, **34**, 859–881.
- Tarantola, A. 1984. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, **49**, 1259–1266.
- Tnacheri, N. O., & Bearnth, R. E. 2007. *Method of seismic interpretation: US Patent 7,519,476*.
- Toldi, J. 1985. *Velocity analysis without picking*. Ph.D. thesis, Stanford University.
- Valenciano, A. 2008. *Imaging by wave-equation inversion*. Ph.D. thesis, Stanford University.
- van Vliet, L. J., & Verbeek, P. W. 1995. Estimators for orientation and anisotropy in digitized images. *Proceedings of the first annual conference of the Advanced School for Computing and Imaging ASCI95, Heijen (The Netherlands)*, 442–450.
- Whitmore, N. D. 1983. Iterative depth migration by backward time propagation. *53rd Annual International Meeting, SEG, Expanded Abstracts*, 382–385.
- Xu, S. 2012. Full waveform inversion for reflected seismic data. *74th EAGE Conference and Exhibition, Extended Abstracts*.

Yan, J., & Tsvankin, I. 2008. AVO-sensitive semblance analysis for wide-azimuth data. *Geophysics*, **73**, U1–U11.

Yilmaz, O. 2001. *Seismic data analysis*. Society of Exploration Geophysicists.