3D seismic image processing for interpretation

Xinming Wu

- Doctoral Thesis -
Geophysics

Defended on 5 February 2016

Advisor: Prof. Dave Hale
Committee Members: Prof. Bernard Bialecki
Prof. Thomas Davis
Prof. Paul Sava
Prof. Gongguo Tang
3D SEISMIC IMAGE PROCESSING
FOR INTERPRETATION

by
Xinning Wu
Extracting fault, unconformity, and horizon surfaces from a seismic image is useful for interpretation of geologic structures and stratigraphic features. Although interpretation of these surfaces has been automated to some extent by others, significant manual effort is still required for extracting each type of these geologic surfaces. I propose methods to automatically extract all the fault, unconformity, and horizon surfaces from a 3D seismic image. To a large degree, these methods just involve image processing or array processing which is achieved by efficiently solving partial differential equations.

For fault interpretation, I propose a linked data structure, which is simpler than triangle or quad meshes, to represent a fault surface. In this simple data structure, each sample of a fault corresponds to exactly one image sample. Using this linked data structure, I extract complete and intersecting fault surfaces without holes from 3D seismic images. I use the same structure in subsequent processing to estimate fault slip vectors. I further propose two methods, using precomputed fault surfaces and slips, to undo faulting in seismic images by simultaneously moving fault blocks and faults themselves.

For unconformity interpretation, I first propose a new method to compute a unconformity likelihood image that highlights both the termination areas and the corresponding parallel unconformities and correlative conformities. I then extract unconformity surfaces from the likelihood image and use these surfaces as constraints to more accurately estimate seismic normal vectors that are discontinuous near the unconformities. Finally, I use the estimated normal vectors and use the unconformities as constraints to compute a flattened image, in which seismic reflectors are all flat and vertical gaps correspond to the unconformities. Horizon extraction is straightforward after computing a map of image flattening; we can first extract horizontal slices in the flattened space and then map these slices back to the original space to obtain the curved seismic horizon surfaces.
The fault and unconformity processing methods above facilitate automatic flattening and horizon extraction by providing an unfaulted image with continuous reflectors across faults and unconformities as constraints for an automatic flattening method. However, human interaction is still desirable for flattening and horizon extraction because of limitations in seismic imaging and computing systems, but the interaction can be enhanced. Instead of picking or tracking horizons one at a time, I propose a method to compute a volume of horizons that honor interpreted constraints, specified as sets of control points in a seismic image. I incorporate the control points with simple constraint preconditioners in the conjugate gradient method used to compute horizons.
# TABLE OF CONTENTS

ABSTRACT ................................................................. iii

LIST OF FIGURES ....................................................... ix

ACKNOWLEDGMENTS ................................................... xv

CHAPTER 1 INTRODUCTION .............................................. 1

1.1 Fault processing ..................................................... 2

1.2 Unconformity processing .......................................... 5

1.3 Image flattening and horizon extraction .......................... 7

1.4 Publications and proceedings ..................................... 9

CHAPTER 2 3D SEISMIC IMAGE PROCESSING FOR FAULTS ............ 11

2.1 Summary .............................................................. 11

2.2 Introduction ......................................................... 11

2.3 Fault images .......................................................... 15

2.4 Fault samples and surfaces ....................................... 18

2.4.1 Fault samples ..................................................... 19

2.4.2 Fault surfaces .................................................... 20

2.4.3 Linking fault sample neighbors ............................... 20

2.4.4 Interpolating missing neighbors .............................. 22

2.5 Fault dip slips ....................................................... 26

2.5.1 Fault throws ...................................................... 26

2.5.2 Unfaulted images ............................................... 29
5.3.4 Horizon extraction with constraints ........................................ 88
5.3.5 Constrained optimization .................................................... 88
5.3.6 Constrained preconditioner ................................................... 89
5.3.7 Results with constraints ..................................................... 91
5.4 Generating a horizon volume ................................................... 93
  5.4.1 Horizon volume without constraints ...................................... 94
  5.4.2 Horizon volume with constraints ......................................... 98
  5.4.3 3D results with constraints ................................................ 102
5.5 Conclusion ............................................................................ 106
5.6 Acknowledgments .................................................................. 107

CHAPTER 6 CONCLUSIONS AND SUGGESTIONS ................................. 108
  6.1 Fault interpretation .............................................................. 108
  6.2 Unconformity interpretation .................................................. 109
  6.3 Image flattening and horizon extraction ................................... 111

REFERENCES CITED .................................................................. 113
## LIST OF FIGURES

| Figure 1.1 | A 3D view of a seismic image (a) and automatically extracted fault (vertical surfaces), unconformity (the two lateral magenta surfaces), and horizon (lateral surfaces) surfaces (b). The unconformities (magenta) and horizons intersect with the seismic inline and crossline slices in (c). | 1 |
| Figure 1.2 | A 3D seismic image with fault likelihoods (a), fault samples (b), and fault surfaces. | 3 |
| Figure 1.3 | The fault surfaces in Figure 1.2c is displayed as a fault likelihood image (a) with mostly null values. This image is used to constrain a structure-oriented filter so that it smoothes along structures, but not across faults, as shown in (b). Fault slip vectors are then estimated using this smoothed image and the extracted fault surfaces. Only fault throws (c), the vertical components of slips are displayed on the fault surfaces. | 3 |
| Figure 1.4 | Fault positions (Figure 1.3a) and fault slip vectors (Figure 1.3c) are used to compute vertical (a), inline (not shown), and crossline (not shown) unfaulting shifts, which undo the faulting in the original seismic image (b) to compute an unfaulted image (c). | 4 |
| Figure 1.5 | Unconformity likelihood displayed in color (a), and two unconformity surfaces (b) and (c) extracted on the ridges of the unconformity likelihoods. | 6 |
| Figure 1.6 | The two unconformity surfaces (Figures 1.5b and 1.5c) extracted from the unfaulted seismic image, are mapped back to the original seismic image using the unfaulting shifts, as shown in (a) and (b), respectively. | 6 |
| Figure 1.7 | The unconformities (Figures 1.5b and 1.5c) are used as constraints to first compute an RGT volume (a) in the unfaulted space, the unfaulted image (b) is then flattened (c) using the RGT volume. | 8 |
| Figure 1.8 | The unfaulting and flattening facilitate extracting any number of seismic horizons from a seismic image. Shown here are only three subsets of the extracted horizon surfaces. | 8 |
| Figure 2.1 | A small subset of 3D seismic image displayed with a fault image (a), fault samples (b), and fault surfaces (c), all colored by fault likelihood. | 12 |
Figure 2.2  Fault dip slip (a) is a vector representing displacement, in the dip direction, of the hanging wall side of a fault surface relative to the footwall side. Fault throw is the vertical component of the slip. Fault strike and dip angles, with corresponding unit vectors are defined in (b). 12

Figure 2.3  A 3D synthetic seismic image (a) with four faults manually interpreted in (b). The dashed lines in (b) represent normal faults, while the solid line represents a reverse fault. 14

Figure 2.4  A 3D seismic image with fault likelihoods (a), strikes (b) and dips (c) displayed in color. The dashed white circle in each image indicates one location where fault F-A intersects fault F-B. 16

Figure 2.5  Thinned fault likelihood image (a) has non-zero values only on the ridges of the fault likelihood image in Figure 2.4a. Fault strikes and dips corresponding to the fault likelihoods are displayed in (b) and (c), respectively. The dashed white circle in each image indicates one location where fault F-A intersects fault F-B. 17

Figure 2.6  The three fault images in Figure 2.5 can be represented by fault samples displayed as small squares (a). Each square in (a) is colored by fault likelihood and oriented by the strike and dip of the corresponding fault sample. Links are then built among consistent fault samples, and each set of linked fault samples in (b) represents a fault surface that appears opaque in (c), where fault samples are displayed as larger overlapping squares. 19

Figure 2.7  Close-up view (a) of a subset of fault samples from Figure 2.6a. Links built among nearby fault samples form three sets of linked samples (b) which represent three fault surfaces (or patches). Near the intersection of faults F-A and F-B, fault F-A is separated into two independent patches, and fault F-B has a hole. New fault samples (colored by yellow and blue) are created to merge the fault patches and fill the hole to construct more complete intersecting fault surfaces (c). 20

Figure 2.8  Linked fault samples in Figure 2.6c can be displayed as a fault likelihood image (with mostly zeros) overlayed with the seismic image in (a). Compared to the thinned fault likelihood image in Figure 2.5a, spurious fault samples have been removed. New fault samples are created at the intersection (dashed white circle) of faults A and B when constructing surfaces. The seismic image in (b) is smoothed along structures, but not across the faults. 25
Figure 2.9  Fault surfaces and fault throws (a) for a 3D seismic image before (b) and after (c) unfaulting. In all image slices, reflectors are more continuous after unfaulting. ................................. 28

Figure 2.10  Fault samples colored by fault likelihood (a) are computed, and linked to form fault surfaces (b). ................................................................. 30

Figure 2.11  Fault surfaces and fault throws for a 3D seismic image before (a) and after (b) unfaulting. In all image slices, reflectors are more continuous after unfaulting. The red arrows point to a large-slip fault before and after unfaulting. ................................................... 31

Figure 3.1  A 3D synthetic seismic image with faults colored by fault throws (a), is significantly distorted when unfaulted (b) by moving only fault blocks while fixing fault positions. Faults (especially fault A) must also be moved to obtain an unfaulted image (c) with minimal distortions. ....... 36

Figure 3.2  Given a 3D seismic image (a), we extract fault surfaces (b) and estimate fault dip slip vectors for each sample on fault surfaces. Faults in (b) and (c) are colored by fault throws, the vertical components of slip vectors. 38

Figure 3.3  A fault slip vector $\mathbf{t}(x_a)$, estimated at each footwall sample adjacent to a fault, tells us how to correlate the image sample at $x_a$ in the footwall to the corresponding sample $x_b$ in the hanging wall. ................................. 41

Figure 3.4  Vertical (a), inline (b) and crossline (c) components of unfaulting shifts $s(x)$ are computed in the input space using method I. Discontinuities in each component of shifts coincide with fault locations. ......................... 44

Figure 3.5  Vertical (a), inline (b) and crossline (c) components of unfaulting shifts $s(x)$ are computed in the input space using method II. Discontinuities in each component of shifts coincide with fault locations. ......................... 44

Figure 3.6  Vertical (a), inline (b) and crossline (c) components of unfaulting shifts in the unfaulted space are converted from those in the input space shown in Figure 3.4. The discontinuities on each component of shifts are displaced relative to those in Figure 3.4. ................................. 47

Figure 3.7  Vertical (a), inline (b) and crossline (c) components of unfaulting shifts in the unfaulted space are converted from those in the input space shown in Figure 3.5. The discontinuities on each component of shifts are displaced relative to those in Figure 3.5. ................................. 47
Figure 3.8 The input synthetic seismic image (a) is unfaulted (b) using shifts in Figure 3.6 computed by method I, and (c) using shifts in Figure 3.7 computed by method II. ................................. 48

Figure 3.9 Fault surfaces and slip vectors (a) are first estimated from a 3D seismic image, and then are used to compute unfaulting vector shifts (b) used below in image unfaulting. Only vertical components of vectors are shown here. ................................. 50

Figure 3.10 A 3D seismic image before (a) and after (b) unfaulting. In all image slices, seismic reflectors are more continuous after unfaulting. For the large-throw fault highlighted by a red arrow in (a), the corresponding fault blocks are significantly moved in (b) to align seismic reflectors on opposite sides of this fault. ................................. 51

Figure 3.11 Composite shifts (a) are computed and then used to obtain an unfaulted and unfolded image (b). ................................. 52

Figure 3.12 Two horizon surfaces (colored by depth) are extracted using composite shift vectors that map an image from input space to unfaulted and unfolded space. The vertical component of the composite shift vectors is displayed in Figure 3.11a. ................................. 53

Figure 4.1 A 2D synthetic seismic image (a) with an unconformity (red curve) that is manually interpreted from the termination area to its corresponding parallel unconformity. The estimated seismic normal vectors (red segments in (b)) are smoothed within the termination area, and therefore are incorrect, compared to the true seismic normal vectors (cyan segments in (b)) that are discontinuous within that area. ................................. 60

Figure 4.2 Two different seismic normal vector fields estimated using structure tensors computed with vertically causal (green segments) and anti-causal (yellow segments) smoothing filters. In (a), the vector fields differ only within the termination area of the unconformity; in (b), these vector differences are extended to the parallel unconformity. ................................. 66

Figure 4.3 Unconformity likelihoods, an attribute that evaluates differences between two estimated seismic normal vector fields (yellow and green segments in Figure 4.2b), before (a) and after (b) thinning highlight both the termination area and parallel unconformity. ................................. 68

Figure 4.4 Unconformity likelihoods before (a) and after (b) thinning. ................................. 68
| Figure 4.5 | Unconformity likelihoods before (a) and after (b) thinning. Thinned unconformity likelihoods form unconformity surfaces as shown in the top-right panel in (b). | 69 |
| Figure 4.6 | Vertical ($u_1$) and horizontal ($u_2$) components of the true (a, d) normal vectors of the synthetic image (Figure 4.1a), the estimated normal vectors (c, f) with the detected unconformity (Figure 4.3b) as constraints are more accurate than those (b, e) without constraints. | 71 |
| Figure 4.7 | From the seismic image as shown in Figure 4.4, the vertical ($u_1$) and horizontal ($u_2$) components of seismic normal vectors estimated using structure tensors computed with (b, d) and without (a, c) unconformity constraints. | 72 |
| Figure 4.8 | RGT (a) and flattened (c) images generated with inaccurate seismic normal vectors (Figures 4.7a and 4.7c) and without unconformity constraints. Improved RGT (b) and flattened (d) images with more accurate seismic normal vectors (Figures 4.7b and 4.7d) and constraints from unconformity likelihoods (Figure 4.4). | 74 |
| Figure 4.9 | Generated RGT volume (a) and flattened (b) 3D seismic image. Discontinuities in the RGT volume correspond to vertical gaps or hiatuses (blank areas in (b)) in the flattened image at unconformities. | 75 |
| Figure 5.1 | From a seismic image (a), an RGT volume (b) is computed and then converted to a horizon volume (c) that maps the seismic image to a flattened image (d). | 80 |
| Figure 5.2 | Seismic sections (a) and subsections (b) that intersect with a sequence boundary. The initially horizontal surface (black curve) passes through one control point and is updated iteratively using seismic normal vectors. The dashed green curve denotes the manually interpreted sequence boundary. | 87 |
| Figure 5.3 | Seismic sections (a) intersect sequence boundaries extracted using one control point (blue curve) and 19 control points (green curve). (b) and (c) show a 3D view of the extracted surfaces using one control point and 19 control points, respectively. Both of the two surfaces are colored by amplitude. | 92 |
| Figure 5.4 | The same RGT volume (a) as shown in Figure 5.1b, contours (b) of the RGT volume are horizons in the corresponding seismic image. | 94 |
| Figure 5.5 | A seismic image (a), generated horizon volume (b), and flattened image (c) without control points. | 97 |
Figure 5.6  A seismic image (a) with 3 pairs of interactively interpreted control points (yellow circles, pluses and squares), generated horizon volume (b) and flattened image (c). .................................................. 102

Figure 5.7  Input seismic image (a) and a corresponding RGT volume (b) computed with three sets of control points. .................................................. 103

Figure 5.8  The flattened seismic image is sliced at \( \tau = 1.664 \) (a) and \( \tau = 1.751 \) (b). Horizontal slices in a flattened image correspond to seismic horizon surfaces (upper-right panels in (a) and (b), for which color denotes depth) in an unflattened image. .................................................. 104

Figure 5.9  The RGT volume shown in Figure 5.7b is converted to a complete horizon volume (a), in which horizontally slicing (h1~h6) at six different RGT values yields six seismic horizons displayed in (b). The cut-away views of these six horizons are shown in (c) to reveal more details. .... 105
ACKNOWLEDGMENTS

People told me that “there are not many Dave Hale in the world”, for me, there is only one Dave. Having a chance to be a student of Dave and work with him, I have to thank another great person Dr. Tom Davis. He kindly encouraged me to also send my PhD application to Dave because he thought my research background was more related to Dave and Dave might be a better advisor for me than him. It has turned out that Tom is correct and Dave is the right advisor to me. I enjoyed every time talking with Dave, his quick thoughts, creative ideas, and valuable suggestions impressed and benefited me much. He taught me how to do research, more importantly, to be an independent thinker and researcher that I am today. He taught me how to write simple, beautiful, and efficient codes. He taught me how to organize a technical paper. He taught me how to efficiently make simple and readable slides. He even taught me how to correctly pronounce a single English word. He bought me a small grammar book “The Elements of Style”, and I carry it everyday in my backpack. What I have learnt from Dave are much more than these above, and the most important thing I got from him is the willing to make anything I already have simpler and better.

I want to thank my minor advisor Dr. Bernard Bialecki. I have been benefited a lot from his mathematic classes to understand and numerically solve the partial different equations that are discussed in this thesis. I appreciate that he was willing to take a lot of time to discuss with me about my research, and these discussions has enriched my thesis. I also want to thank all my teachers, especially the geophysics ones, Dr. Dave Hale, Dr. Paul Sava, Dr. Yaoguo Li, and Dr. Ilya Tsvankin. The geophysical knowledge I learnt from them contributed a lot to my research and thesis.

I want to thank all the faculty of Center for Wave Phenomena (CWP) Dr. Dave Hale, Dr. Paul Sava, Dr. Roel Snieder, and Dr. Ilya Tsvankin for their discussions, assistances, and advices. They all are excellent researchers and scientists. Being working with them is a
wonderful experience for shaping me to be a good researcher as them. I want to thank the CWP students for sharing me with their diverse research ideas, experiences, and cultures. I gained new knowledge and happiness by simply being together with them. I also want to thank the CWP staff Diane Witters, Shingo Sean Ishida, Pamela Kraus, and John Stockwell for their assistance so that I could focus more on my research.

I want to thank all the people I met when I interned in Transform which is now a part of DrillingInfo. They generously shared me with their interests and ideas regarding to seismic interpretation. I want to especially thank Dr. Dean Witte, Stefan Compton, John Neave and Bob Howard. All of them are great programmers and expects on seismic interpretation, and I was inspired and encouraged by them to continue working on my research that are presented in this thesis.

I also want to thank Dr. Sergey Fomel for providing me a great chance to visit the group of Texas Consortium for Computational Seismology (TCCS) and work with people in the Bureau of Economic Geology for six months. It was a pleasant and valuable experience to talk and work with the TCCS students who focused on totally different research topics. Discussions with Dr. Sergey Fomel helped me find possibilities to go beyond my current research presented in this thesis and figure out valuable research projects in the future.

Next but not last, I want to thank my family and friends for their continued support. I want to especially thank my wife Rong for the sacrifice of her study and work in order to better take care of the family. Without her assistance, I do not know how can I finish my PhD program if I need to keep watching our baby girl and keep changing her diapers. Our baby Grace has turned out to be not just a gift or source of happiness for me, she actually keeps training me to be a more efficient worker and sleeper by increasingly decreasing my time for both working and sleeping.

Finally, I want to thank all my thesis committee members Dr. Bernard Bialecki, Dr. Paul Sava, Dr. Tom Davis, Dr. Gongguo Tang, and Dr. Andrzej Szymczak for the constructive discussions with them and their revision of my thesis.
CHAPTER 1
INTRODUCTION

From a 3D seismic image, as shown in Figure 1.1a, one can extract geologic surfaces such as faults (like these vertical surfaces in Figure 1.1b), unconformities (like the two magenta surfaces in Figure 1.1b and magenta curves in Figure 1.1c), and horizons (like these lateral surfaces in Figure 1.1b and curves in Figure 1.1c). Faults and horizons are important for seismic structural interpretation because they provide structural maps of the subsurfaces. Horizons and unconformities are important for seismic stratigraphic interpretation because they together construct a chronostratigraphic framework. All these geologic surfaces can be useful for seismic lithology interpretation because they can provide geologically reasonable control for extending a lithology interpretation away from wells. Therefore, extracting these surfaces is a critical part of seismic interpretation.

Numerous automatic methods have been proposed to extract all the three types of surfaces, however, interpreting any of them today typically requires significant manual effort, suggesting that further improvements in automatic methods are feasible and worthwhile.
Moreover, horizons can be especially difficult to extract from a seismic image complicated by both faults and unconformities, as shown in Figure 1.1, because a horizon surface can be dislocated at faults and terminated at unconformities. In this case, we want to first extract fault surfaces, estimate fault slips, and undo the faulting in the seismic image, then to extract unconformities from the unfaulted image with continuous reflectors across faults, finally to use the unconformities as boundary control for image flattening and horizon extraction. In this thesis, I propose 3D seismic image processing methods for (1) fault processing including automatic computation of fault positions, fault slips, and unfaulted images; (2) unconformity processing including automatic extraction of unconformities and estimation of seismic normal vectors at unconformities; (3) image flattening including automatic flattening with constraints from unconformities and semi-automatic flattening with control points or surfaces manually interpreted in the seismic image.

1.1 Fault processing

Automatic interpretation of faults from a seismic image often includes four parts: (1) Fault images or attributes such as semblance (Marfurt et al., 1998), coherency (Marfurt et al., 1999), variance (Randen et al., 2001; Van Bemmel and Pepper, 2000), and fault likelihood (Hale, 2013b) are computed from a seismic image to highlight out fault positions. (2) Then fault surfaces are extracted from these computed fault images using various methods (e.g., Gibson et al., 2005; Hale, 2013b; Pedersen et al., 2002, 2003). (3) From extracted fault surfaces, fault slips are estimated by correlating seismic horizons (Admasu, 2008) or reflectors (Hale, 2013b) on opposite sides of fault surfaces. (4) Computed fault positions and fault slips are finally used to undo the faulting in the seismic image (Luo and Hale, 2013; Wei, 2009; Wei et al., 2005). Although various methods have been proposed for the four parts, the problem of extracting intersecting faults is not well addressed. In addition, extracted fault surfaces are often represented by triangle or quad meshes, which are often more complex than necessary for subsequent processing. Moreover, all the unfaulting methods assume that fault geometries need not change when unfaulting a seismic image. This assumption
makes the unfaulting processing easier, however, often results in unnecessary distortions when unfaulting seismic images with multiple faults and, especially, intersecting faults.

Figure 1.2: A 3D seismic image with fault likelihoods (a), fault samples (b), and fault surfaces.

Figure 1.3: The fault surfaces in Figure 1.2c is displayed as a fault likelihood image (a) with mostly null values. This image is used to constrain a structure-oriented filter so that it smoothes along structures, but not across faults, as shown in (b). Fault slip vectors are then estimated using this smoothed image and the extracted fault surfaces. Only fault throws (c), the vertical components of slips are displayed on the fault surfaces.

In **Chapter 2**, I first describe how to compute images of fault likelihood (Figure 1.2a), strike (not shown), and dip (not shown), and then represent these three images, all at once, by fault samples as shown in Figure 1.2b. Each fault sample corresponds to one and only one seismic image sample, and is displayed as a small square colored by fault likelihood and oriented by strike and dip. I then propose a method to link these oriented fault samples to construct complete fault surfaces without holes, as shown in Figure 1.2c. These surfaces
are really just sets of linked fault samples, they appear as opaque surfaces because fault samples are represented with larger and overlapping squares in Figure 1.2c. These colored fault surfaces can be displayed as a 3D fault likelihood image, with mostly null values, overlaid with the seismic image in Figure 1.3a. This fault image is used to constrain a structure-oriented filter (Fehmers and Höcker, 2003; Hale, 2009) so that it smoothes along structures, but not across faults, to obtain the smoothed seismic image shown in Figure 1.3b. This smoothing does what seismic interpreters do visually when estimating fault slips, by bringing seismic amplitudes from within each fault block up to, but not across, the faults. Using this smoothed image and the extracted fault surfaces without holes, I finally estimate fault slips, and display fault throws, the vertical components of the slips, on the fault surfaces shown in Figure 1.3c.

Figure 1.4: Fault positions (Figure 1.3a) and fault slip vectors (Figure 1.3c) are used to compute vertical (a), inline (not shown), and crossline (not shown) unfaulting shifts, which undo the faulting in the original seismic image (b) to compute an unfaulted image (c).

In Chapter 3, I introduce two methods to compute vertical (Figure 1.4a), inline (not shown), and crossline (not shown) unfaulting shifts for all samples in a seismic image by solving simple equations derived from the precomputed fault positions and fault slip vectors. These computed vector shifts simultaneously move footwalls, hanging walls, and even the faults themselves, to undo faulting in a seismic image, with minimal distortion as shown in Figure 1.4c.
1.2 Unconformity processing

Unconformity extraction from seismic images is important for seismic stratigraphic interpretation, because unconformities represent discontinuities in otherwise continuous deposits and hence serve as boundaries when interpreting seismic sequences that represent successively deposited layers. To obtain complete unconformities from a seismic image, we want to extract both angular unconformities with reflector terminations and the corresponding parallel unconformity or correlative conformity with conformable reflectors. Most automatic methods (e.g., Bahorich and Farmer, 1995; Barnes et al., 2000; Hoek et al., 2010; Smythe et al., 2004) can only detect angular unconformities by computing different kinds of attributes that highlight areas of reflector terminations.

In Chapter 4, I propose a method to compute an unconformity likelihood image that highlight both termination areas and the corresponding parallel unconformities as shown in Figure 1.5a. In this method, the unconformity likelihood is defined as the difference between two seismic normal vector fields corresponding to two structure tensor fields constructed from a same seismic image using different smoothing filters. One structure-tensor field is constructed by applying a laterally structure-oriented smoothing filter and a vertical causal filter to each element of the outer products of seismic image gradients. The other one is constructed by applying a same laterally structure-oriented smoothing filter but a vertical anticausal filter. Near the termination area of an unconformity, the reflector structures above and below the unconformity must be different. Therefore, the vertical causal filter, which computes locally averaged structures from above of the unconformity, yields a structure-tensor field that is different from the one constructed with the vertical anticausal filter, which computes locally averaged structures from below. The lateral structure-oriented smoothing filter extends the structure differences, which originate within the termination areas, to the corresponding parallel unconformities and correlative conformities.

Using these lateral and vertical smoothing filters, the two constructed structure-tensors fields and the two corresponding vector fields should be different at both termination area
and the corresponding parallel unconformity and correlative conformity. Therefore, the unconformity likelihoods, defined as the differences between the two vector fields, should be relatively high at both the angular unconformities and the corresponding parallel unconformities and correlative conformities as shown in Figure 1.5. From the unconformity likelihood image, two unconformity surfaces (Figures 1.5b and 1.5c) are extracted on the ridges of the unconformity likelihoods.

This method assumes that unconformities are not dislocated by faults, so that the lateral structure-oriented smoothing filter can extend structure differences from termination areas to the corresponding parallel unconformities and correlative conformities. Therefore, if faults
appear in the seismic image, we should perform unfaulting before attempting to detect unconformities, as in this example shown in Figure 1.5. These unconformity surfaces shown in Figures 1.5b and 1.5c, extracted from the unfaulted image, are then mapped back to the original seismic image, as shown in Figures 1.6a and 1.6b, respectively. These faulted unconformities are difficult to extract directly from the original seismic image for any automatic or manual methods.

As applications in Chapter 4, I first use these extracted unconformity surfaces as constraints for a structure-tensor method (Fehmers and Höcker, 2003; Van Vliet and Verbeek, 1995) to more accurately estimate seismic normal vectors at unconformities with multioriented seismic reflectors. I then use the more accurate seismic normal vectors, and unconformity surfaces as constraints, to compute a flattened image with all flat seismic reflectors within conformable areas and vertical gaps corresponding to the unconformities.

1.3 Image flattening and horizon extraction

Automatic seismic image flattening (Lomask et al., 2006; Parks, 2010; Stark, 2005; Wu and Zhong, 2012a) is a volume process method to identify all horizons in a seismic image by computing a map that transforms the seismic image from the original space into the flattened space. This map can be used to extract any number of horizons from the seismic image. These methods, however, are unable to match horizons across faults unless additional information such as fault slips (Luo and Hale, 2013) and control points across faults (Wu and Hale, 2015b) are provided; also are difficult to deal with horizons terminated at unconformities unless the unconformity surfaces are provided (Wu and Hale, 2015a; Wu and Zhong, 2012a).

For a seismic image complicated only by faults, we could first use the automatic fault processing methods discussed in Chapter 2 and Chapter 3 to estimate fault slips and compute an unfaulted image. Then we could use any flattening method discussed above to compute a flattened or unfolded image from the unfaulted image with continuous reflectors across faults. For a seismic image complicated by both faults and unconformities, as shown in
Figure 1.7: The unconformities (Figures 1.5b and 1.5c) are used as constraints to first compute an RGT volume (a) in the unfaulted space, the unfaulted image (b) is then flattened (c) using the RGT volume.

Figure 1.8: The unfaulting and flattening facilitate extracting any number of seismic horizons from a seismic image. Shown here are only three subsets of the extracted horizon surfaces.

Figure 1.1a, we still first undo the faulting in the seismic image, as shown in Figure 1.4. Then we extract unconformities from the unfaulted image, as shown in Figure 1.5. Finally we could use the extracted unconformities as constraints, as discussed in Chapter 4, to compute a relative geologic time (RGT) volume shown in Figure 1.7a. This RGT volume, with discontinuous values at unconformities, is applied to the unfaulted image (Figure 1.7b) to compute a flattened image with vertical gaps corresponding the unconformities, as shown in Figure 1.7c.

Horizon extraction is trivial after computing the maps of unfaulting and flattening. For example, I first extract horizontal slices in the unfaulted and flattened space (Figure 1.7c), I then use the RGT volume (Figure 1.7a) to map these horizontal slices back to the unfaulted
space (Figure 1.7b) and obtain curved surfaces, I finally use the unfaulting shifts (Figure 1.4b) to map the curved surfaces back to the original space and eventually obtain curved and faulted surfaces as shown in Figure 1.8. In this way, I can easily compute any number of seismic horizons in the original space. Figure 1.8 shows only three subsets of seismic horizons, which are curved and faulted.

Although all of this image processing can be performed automatically to compute a volume of seismic horizons, limitations inherent in seismic imaging and computing systems suggest that human interaction will continue to be desirable. In Chapter 5, I propose an enhanced semi-automatic method to compute a horizon volume that honors both the seismic image and human interactions. In this method, instead of picking or tracking horizons one at a time as usual, I interactively select scattered sets of points in a 3D seismic image that correspond to multiple horizons, while automatically updating a complete 3D horizon volume to honor those interpreted constraints.

1.4 Publications and proceedings

The work discussed in Chapters 1, 2, 3, 4, and 5 of this thesis have been published in, or submitted to the journals. Below is a full list of my publications:


In addition to the publications above, I have also contributed several conference abstracts:


2.1 Summary

Numerous methods have been proposed to automatically extract fault surfaces from 3D seismic images, and those surfaces are often represented by meshes of triangles or quadrilaterals. Such mesh data structures are more complex than the arrays used to represent seismic images, and are more complex than necessary for subsequent processing tasks, such as that of automatically estimating fault slip vectors. To facilitate image processing for faults, we propose a simpler linked data structure in which each sample of a fault corresponds to exactly one image sample. Using this linked data structure, we extracted multiple intersecting fault surfaces from 3D seismic images. We then used the same structure in subsequent processing to estimate fault slip vectors, and to assess the accuracy of estimated slips by unfaulting the seismic images.

2.2 Introduction

Faults like those shown in Figure 3.1 are important geologic surfaces that we can automatically extract from 3D seismic images. When extracting a fault surface, we also want to obtain fault strikes, dips, and slip vectors, as illustrated in Figure 2.2.

To extract fault surfaces, fault images (like that shown in Figure 3.1a) are first computed from a seismic image. These fault images indicate where faults might exist. Many methods have been developed to compute fault images using attributes such as semblance (Marfurt et al., 1998), coherency (Marfurt et al., 1999), variance (Randen et al., 2001; Van Bemmel

\footnote{Center for Wave Phenomena, Department of Geophysics, Colorado School of Mines}
Figure 2.1: A small subset of 3D seismic image displayed with a fault image (a), fault samples (b), and fault surfaces (c), all colored by fault likelihood.

Figure 2.2: Fault dip slip (a) is a vector representing displacement, in the dip direction, of the hanging wall side of a fault surface relative to the footwall side. Fault throw is the vertical component of the slip. Fault strike and dip angles, with corresponding unit vectors are defined in (b).

and Pepper, 2000), gradient magnitude (Aqrawi and Boe, 2011) and fault likelihood (Hale, 2013b).

Various methods are also proposed to extract fault surfaces from computed fault images, Pedersen et al. (2002) and Pedersen et al. (2003) propose the ant tracking method to first extract small fault segments, which are then merged to form larger fault surfaces. Similarly, the methods, proposed by Gibson et al. (2005), Admasu et al. (2006) and Kadlec et al. (2008), also try to build larger fault surfaces from smaller patches. Hale (2013b) uses images of fault likelihoods, strikes and dips to construct fault surfaces that coincide with ridges of the fault likelihood image.
From extracted fault surfaces, fault slips can be estimated by correlating seismic reflectors or horizons on opposite sides of the fault surfaces. For example, Borgos et al. (2003) correlate seismic horizons across faults by using a clustering method with multiple seismic attributes. Admasu (2008) propose to use a Bayesian matching of seismic horizons extracted on opposite sides of faults. Aurnhammer and Tonnies (2005) and Liang et al. (2010) use windowed crosscorrelation methods to correlate seismic reflectors across faults. Hale (2013b) uses a dynamic image warping method.

As discussed above, various methods have been proposed to compute fault images, extract fault surfaces and estimate fault slips. However, the problem of extracting intersecting faults, like those shown in Figure 3.1, is not well addressed. For example, the method described by Hale (2013b) assumes that a single seismic image sample can be associated with only one fault, and therefore extracts incomplete fault surfaces, with holes at intersections. Incomplete fault surfaces cause problems for all the above methods used to estimate fault slips, because near holes it is difficult to determine which seismic reflectors should be correlated.

This paper contributes mainly to two aspects of image processing for faults: Firstly, we address the problem of extracting intersecting faults, and obtain complete fault surfaces without holes. Secondly, we propose to represent a fault surface using a linked data structure that is simpler than triangle or quadrilateral meshes often used for fault surfaces.

We first use the method described in (Hale, 2013b) to compute images of fault likelihoods, strikes and dips. Each of the these images has non-null values only at faults, as in the fault likelihood image shown in Figure 3.1a. Therefore, these three fault images can be represented, all at once, by fault samples shown in Figure 3.1b. Each fault sample corresponds to one and only one seismic image sample, and can be displayed as a small square colored by fault likelihood and oriented by strike and dip.

We then use the fault samples in Figure 3.1b to construct fault surfaces, which appear to be continuous, as shown in Figure 3.1c, but are actually only linked lists of the fault samples in Figure 3.1b. In Figure 3.1c, we simply increase the size of squares that are used to
represent fault samples, so that they overlap and appear to form continuous surfaces. Each of these fault surfaces is constructed by linking each fault sample with neighbors above, below, left and right. If any of the four neighbors are missing, we attempt to create them using a method proposed in this paper. In this way, we fill holes and merge separated fault segments to form more complete and intersecting faults as shown in Figure 3.1c.

With more complete surfaces without holes, we are able to more accurately estimate fault slips. To verify the estimated slips, we use them in an unfaulting processing to correlate seismic reflectors across faults.

Figure 2.3: A 3D synthetic seismic image (a) with four faults manually interpreted in (b). The dashed lines in (b) represent normal faults, while the solid line represents a reverse fault.
2.3 Fault images

To illustrate our 3D seismic image processing for: (1) computing images of fault likelihood, strike and dip, (2) constructing fault samples from thinned fault images, (3) linking fault samples to form fault surfaces, and (4) estimating fault dip slip vectors, we created a synthetic 3D seismic image with normal, reverse, and intersecting faults, as shown in Figure 2.3. This synthetic image contains two intersecting normal faults F-A and F-B, a reverse fault F-C, and a smaller normal fault F-D. While somewhat unrealistic, this synthetic image provides a good test for processing of both normal and reverse faults, and intersecting faults.

In 3D seismic images like those shown in Figures 3.1, and 2.3, faults appear as discontinuities that are locally planar (or locally linear in 2D slices). This means that, to highlight faults from a seismic image, we do not only look for discontinuities, but rather for discontinuities that are locally planar. Fault likelihood, as defined by Hale (2013b), is one such measure of locally planar discontinuity. Therefore, we use Hale’s (2013b) method to compute fault likelihood (Figure 2.4a), while at the same time estimating fault strike (Figure 2.4b) and dip (Figure 2.4c). The fault likelihood image indicates where faults might exist, while the strike and dip images indicate their orientations.

In computing these fault images, this method scans over a range of possible combinations of strike and dip to find the one orientation that maximizes the fault likelihood, for each image sample (Hale, 2013b). The maximum fault likelihood for each image sample is recorded in the fault likelihood image (Figure 2.4a), and the strike and dip angles that yield the maximum likelihood are recorded in the fault strike (Figure 2.4b) and dip (Figure 2.4c) images, respectively. In the fault likelihood image in Figure 2.4a, we expect relatively high values in areas where faults might exist. In the fault strike (Figure 2.4b) and dip (Figure 2.4c) images, we expect strike and dip angles to be accurate only where faults are likely, that is, where fault likelihoods are high.

As discussed by Hale (2013b), a significant limitation of this scanning method is in dealing with intersecting faults. Because only a single fault likelihood value, and its corresponding
Figure 2.4: A 3D seismic image with fault likelihoods (a), strikes (b) and dips (c) displayed in color. The dashed white circle in each image indicates one location where fault F-A intersects fault F-B.
Figure 2.5: Thinned fault likelihood image (a) has non-zero values only on the ridges of the fault likelihood image in Figure 2.4a. Fault strikes and dips corresponding to the fault likelihoods are displayed in (b) and (c), respectively. The dashed white circle in each image indicates one location where fault F-A intersects fault F-B.
fault strike and dip are recorded for each image sample, this method implicitly assumes that each image sample can be associated with only one fault. This assumption is not valid for samples where two or more faults intersect. For example, in the intersection area highlighted in Figure 2.4, fault likelihoods, strikes and dips for only fault F-B have been recorded, and the information corresponding to fault F-A is missing there. Fault likelihoods of fault F-A might also be high near this intersection, but have been discarded together with corresponding strikes and dips, only because they were smaller than the fault likelihoods computed for fault F-B. Therefore, fault surfaces directly extracted from such fault images often have holes, especially near fault intersections. We describe below a method to fill holes when constructing fault surfaces.

We do not expect faults to be as thick as the features apparent in the fault likelihood image in Figure 2.4a. Therefore, we keep only the values on the ridges of fault likelihood, and set values elsewhere to be zero, to obtain a thinned fault likelihood image shown in Figure 2.5a. We also keep strike and dip angles for only the samples with non-zero values in Figure 2.5a, to obtain the corresponding thinned fault strike (Figure 2.5b) and dip (Figure 2.5c) images. These thinned fault images have non-null values only for samples that might be on faults.

We again observe that fault likelihoods, strikes and dips of fault F-A are missing in the intersection area (dashed white circles in Figure 2.5) because of the limitation discussed above. We also observe that some non-null values appear where faults do not exist. The reason for this is that, in the scanning process, we assume only that faults are locally planar discontinuities. We will discuss below additional conditions that must be satisfied for the non-null samples in Figure 2.5 to be considered as faults.

2.4 Fault samples and surfaces

Notice that most samples in thinned fault images shown in Figure 2.5 are null. For this reason, we use fault samples to more efficiently represent the three fault images with less computer memory. We then extract fault surfaces from the fault samples, and represent the
surfaces using simple and convenient data structures.

2.4.1 Fault samples

Because most samples in the images of fault likelihood (Figure 2.5a), strike (Figure 2.5b) and dip (Figure 2.5c) are null, we can display the three images, all at once, as fault samples shown in Figure 2.6a, and more clearly in Figure 2.7a. Each fault sample is displayed as a colored square. The color of each square denotes fault likelihood, while the orientation of each square represents fault strike and dip. Fault samples exist only at positions where thinned fault likelihoods are non-null, and each fault sample corresponds to one and only one image sample. Therefore, these fault samples contain exactly the same information represented in the thinned fault images shown in Figure 2.5.

Most fault samples, especially those with high fault likelihoods, are aligned approximate planes, consistent with locally planar fault surfaces. Some misaligned fault samples, often with low fault likelihoods, are also observed in Figures 2.6 and 2.7. These misaligned samples, however, cannot be linked together to form locally planar fault surfaces of significant size.

Figure 2.6: The three fault images in Figure 2.5 can be represented by fault samples displayed as small squares (a). Each square in (a) is colored by fault likelihood and oriented by the strike and dip of the corresponding fault sample. Links are then built among consistent fault samples, and each set of linked fault samples in (b) represents a fault surface that appears opaque in (c), where fault samples are displayed as larger overlapping squares.
2.4.2 Fault surfaces

Fault surfaces are often represented by meshes of triangles or quadrilaterals (Hale, 2013b). However, these mesh structures are unnecessary for the image processing described in this paper.

For example, to estimate fault slips, we must analyze seismic image samples alongside faults. This means that we must know how to walk vertically up and down (tangent to fault dip), and horizontally left and right (tangent to fault strike), on a fault surface, and thereby to access seismic image samples adjacent to the fault. The quadrilateral mesh discussed by Hale (2013b) is one way to efficiently access seismic image samples alongside a fault. In this paper, we use a simpler linked data structure, shown in Figures 2.6b, 2.7b and 2.7c, to find seismic image samples adjacent to faults.

2.4.3 Linking fault sample neighbors

To link fault samples into fault surfaces, we use their fault likelihoods, strikes and dips. We grow a fault surface by linking nearby fault samples with similar fault attributes, begin-
ning with a seed sample that has sufficiently high fault likelihood. Remember that each fault sample corresponds to exactly one sample of the seismic image. This means that we can use the image sampling grid to efficiently search for neighbor samples that should be linked. In a 3D sampling grid, each fault sample has 26 adjacent grid points in a $3 \times 3 \times 3$ cube centered at that sample, but most of these adjacent grid points will not have a fault sample. At these adjacent grid points, we search for up to four neighbor fault samples, above and below (in directions best aligned with fault dip), left and right (in directions best aligned with fault strike).

To find a neighbor above, we need only consider the upper 9 adjacent points in the $3 \times 3 \times 3$ cube of grid points. Among these 9 grid points, we search for a fault sample that lies nearest to the line defined by the center fault sample and its dip vector. Similarly, we search for a neighbor below among the lower 9 adjacent grid points.

To find a neighbor right and left, we need only search in the 8 adjacent grid points with the same depth as the center fault sample in the $3 \times 3 \times 3$ cube. The right neighbor is the one located in the strike direction and nearest to the line defined by the center fault sample and its strike vector. The left neighbor is the one in the opposite direction and closest to the same line.

The fault samples (up to four) obtained in this way are only candidate neighbors. To be considered as valid neighbors and then linked to the center sample, they must have fault likelihoods, dips and strikes similar to those for the center sample.

The processing above is repeated for each fault sample neighbor until no more neighbors can be found, to obtain a linked list of fault samples (Figure 2.7b). Then a new seed with sufficiently high fault likelihood is chosen from unlinked samples for growing a new fault surface. This process ends when no remaining unlinked fault samples have sufficiently high fault likelihood.

Some samples linked in this way may not correspond to faults. We discard surfaces with small numbers of linked samples, and keep only those with significant sizes. For an
example, in Figure 2.7b, we have kept only the three largest surfaces constructed from the
fault samples in Figure 2.7a. Other fault samples (such as these colored by green and blue
in Figure 2.7a) are then ignored in subsequent processing.

As shown in Figure 2.7b, each sample in a fault surface is linked to up to four neighbors.
Some neighbors might be missing, and this is in fact necessary, because faults are not per-
factly aligned with the sampling grid of a 3D seismic image, and also because faults are not
strictly planar surfaces.

However, some neighbors may be missing because the seismic image is noisy. And where
faults intersect, fault samples constructed directly from fault images may be missing, as
shown in Figure 2.7a. These missing fault samples can cause holes within a fault surface,
like the fault F-B shown in Figure 2.7b, and can yield gaps which separate a fault surface into
independent patches, like those of fault F-A shown in Figure 2.7b. To fill in these holes and
gaps to construct more complete fault surfaces, we must interpolate missing fault samples,
as shown in Figure 2.7c.

2.4.4 Interpolating missing neighbors

During the processing discussed above for linking neighbors to a fault sample, if any of
the neighbors above, below, left or right are missing, we try to create them. We do not first
construct fault surfaces or patches with holes (missing neighbors) as shown in Figure 2.7b,
and then fill holes in each of the constructed fault surfaces or patches, because in this way
we cannot merge fault patches to form more complete fault surface. Instead, we check for
missing neighbors, and create them as we grow fault surfaces, and thereby directly obtain
complete fault surfaces without holes as shown in Figure 2.7c.

Remember that each fault sample contains three attributes: fault likelihood, strike and
dip. This means that, if we want to create a missing fault sample, we must know not only its
position, but also its corresponding fault attributes. Therefore, instead of directly creating
fault samples, we first construct three fault images and then create fault samples from these
images. Recall that we find neighbors for a fault sample from only the adjacent grid points
in a $3 \times 3 \times 3$ cube that is centered at that fault sample. This means that, to create a missing neighbor, we need only create adjacent fault samples, and then determine whether any of them could be the missing neighbor.

To create fault samples within a $3 \times 3 \times 3$ cube, we must create fault images in a slightly larger cube, because fault samples are located on the ridges in a fault likelihood image, and additional image samples are needed to find these ridges. Therefore, we construct the new small images of fault likelihood, strike and dip in a $5 \times 5 \times 5$ cube.

To construct a fault likelihood image in a $5 \times 5 \times 5$ cube centered at the fault sample with missing neighbors, we first search nearby to find fault samples that have fault attributes similar to those for the center sample. For all examples in this paper, we search for these nearby samples in a $31 \times 31 \times 31$ cube. Suppose that we find $N$ existing fault samples, we then construct a $5 \times 5 \times 5$ fault likelihood image by accumulating weighted anisotropic Gaussian functions generated from the $N$ existing fault samples found nearby:

$$f(x_i) = \sum_{k=1}^{N} f(x_k)g(x_k - x_i),$$

(2.1)

where $f(x_i)$ denotes a fault likelihood value computed for the $i$-th grid point in the $5 \times 5 \times 5$ cube, and $x_i$ denotes the position of that grid point. Here, $f(x_k)$ denotes the known fault likelihood of the $k$-th nearby fault sample, and $g(x)$ is an anisotropic Gaussian function computed for this $k$-th fault sample:

$$g(x) = \exp(-\frac{1}{2}x^\top R^\top S Rx).$$

(2.2)

Here, $R$ and $S$ are $3 \times 3$ matrices:

$$R = \begin{bmatrix} u_k^\top \\ v_k^\top \\ w_k^\top \end{bmatrix}, \text{ and, } S = \begin{bmatrix} \frac{1}{\sigma_u^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_v^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_w^2} \end{bmatrix},$$

(2.3)

where the unit column vectors $u_k$ and $v_k$ are the dip and strike vectors of the $k$-th nearby fault sample, respectively. The vector $w_k = u_k \times v_k$ is normal to the plane of the $k$-th fault sample, and $\sigma_u$, $\sigma_v$, and $\sigma_w$ are specified half-widths of the Gaussian function in the dip ($u$),
strike (v) and normal (w) directions, respectively. The matrix R rotates the anisotropic Gaussian to be aligned with the vectors u_k, v_k and w_k. Because a fault should be locally planar in strike and dip directions, we set the half-widths \( \sigma_u \) and \( \sigma_v \) to be larger than \( \sigma_w \), so that the Gaussian to be accumulated extends primarily in the fault strike and dip directions. For all examples in this paper, we set \( \sigma_w = 1 \) and \( \sigma_u = \sigma_v = 15 \) samples.

To create fault samples located on ridges of a fault likelihood image, we must also construct images of fault strikes and dips. Therefore, when accumulating anisotropic Gaussian functions for the \( i \)-th sample in the \( 5 \times 5 \times 5 \) cube, we also accumulate weighted outer products of normal vectors for that sample:

\[
D(x_i) = \sum_{k=1}^{N} f(x_k)g(x_k - x_i)w_kw_k^T.
\] (2.4)

We then apply eigen-decomposition to the \( 3 \times 3 \) matrix \( D(x_i) \), and choose the eigenvector corresponding to the largest eigenvalue to be the normal vector \( w_i \) for the \( i \)-th sample in the \( 5 \times 5 \times 5 \) cube. From each normal vector \( w_i \), we then compute strike and dip angles for this \( i \)-th sample.

After construct the three \( 5 \times 5 \times 5 \) fault images, we then create fault samples on the ridges of the fault likelihood image, and search for missing neighbors among these new fault samples. Again, the conditions discussed in the previous section must be satisfied when finding neighbors from these new samples. If no valid neighbors can be found, we stop linking neighbors to the center fault sample.

Figure 2.7c shows new fault samples created in this way, and colored by yellow and blue, for two different fault surfaces. Using both newly created and original fault samples, we are able to construct intersecting fault surfaces without holes, as shown in Figure 2.7c.

Figure 2.6b shows the four fault surfaces extracted from the 3D seismic image by using the method discussed above. They can be displayed as opaque fault surfaces, as in Figure 2.6c, by simply increasing the size of each square so that they overlap and appear to form continuous surfaces.
Figure 2.8: Linked fault samples in Figure 2.6c can be displayed as a fault likelihood image (with mostly zeros) overlayed with the seismic image in (a). Compared to the thinned fault likelihood image in Figure 2.5a, spurious fault samples have been removed. New fault samples are created at the intersection (dashed white circle) of faults A and B when constructing surfaces. The seismic image in (b) is smoothed along structures, but not across the faults.

However, these surfaces are really just linked fault samples. As shown in Figure 2.7c, the samples on a surface are linked above and below in the fault dip direction, left and right in the strike direction, and no holes are apparent. These links enable us to iterate among seismic image samples adjacent to a fault, in both dip and strike directions, as we estimate fault slips.

Recall that each sample in a fault surface corresponds to exactly one sample of the seismic image, therefore, fault likelihoods for the surfaces shown in Figure 2.6c can be easily
displayed as a 3D fault likelihood image (with non-null values only at faults) overlayed with the seismic image in Figure 2.8a. Compared to the thinned fault likelihood image in Figure 2.5a, spurious fault samples have been removed and new fault samples have been created near the intersection (dashed white circle) of faults A and B.

We create this fault likelihood image to constrain a structure-oriented filter (Fehmers and Höcker, 2003; Hale, 2009) so that it smoothes along structures, but not across faults, to obtain the smoothed seismic image shown in Figure 2.8b. We use this smoothed image in the next section for estimating fault slips, because the smoothing does what seismic interpreters do visually when estimating fault slips, by bringing seismic amplitudes from within each fault block up to, but not across, the faults.

2.5 Fault dip slips

In a 3D seismic image, fault strike slips are typically less apparent than dip slips. Therefore, we have not attempted to estimate fault slips in the strike direction. As shown in Figure 2.2a, fault dip slip is a vector representing displacement, in the dip direction, of the hanging wall side of a fault surface relative to the footwall side. Fault throw is the vertical component of slip. If we know the fault throw and the fault surface with linked samples, as in Figure 2.7c, then we can walk up or down the fault in the dip direction to compute the two corresponding horizontal components of dip slip. Therefore, to estimate dip slips, we first estimate fault throws.

2.5.1 Fault throws

To estimate fault throws, we compute vertical components of shifts that correlate seismic reflectors on the footwall and hanging wall sides of a fault surface. As discussed by Hale (2013b), this correlation can be difficult. The drawing of a fault in Figure 2.2a is a very simple case, where the fault surface is entirely planar, and fault throw is constant for the entire surface. In reality, fault throws may vary significantly within a fault surface, and this variation can make windowed crosscorrelation methods (Aurnhammer and Tonnies, 2005;
Liang et al., 2010) fail when fault throws vary within a chosen window size.

To avoid choosing windows in which throws are assumed to be constant, we use the dynamic image warping method (Hale, 2013a,b) to estimate fault throws. Compared to windowed crosscorrelation methods, dynamic image warping is more accurate, especially when the relative shifts between two images vary rapidly. Moreover, this method enables us to impose constraints on the smoothness of estimated shifts. These constraints are important in fault throw estimation, because we expect throws to vary smoothly and continuously along a fault, even where they may increase or decrease rapidly.

The dynamic image warping method described in Hale (2013a) cannot be used directly to estimate fault throws. This method assumes images to be warped (aligned) are regularly sampled. In practice, a fault is generally not planar; instead, it is often curved and sometimes cannot be projected onto a plane (e.g., Walsh et al., 1999). Therefore, images extracted from opposite sides of a fault are not regularly sampled 2D images as required for dynamic image warping. The dynamic warping method must be modified for fault throw estimation.

Hale (2013b) represents a fault surface as a quad mesh that facilitates computation of differences between seismic amplitudes on opposite sides of a fault. Those amplitude differences are computed for every sample on the fault, for a range of shifts that correspond to different fault throws. The fault throws computed by dynamic warping are shifts that minimize these seismic amplitude differences, subject to constraints that those shifts must vary smoothly along the fault surface.

In this paper, we use the same dynamic warping process, but with seismic amplitude differences computed using the simpler linked data structure illustrated in Figure 2.7c. It is advantageous that the fault surfaces represented in Figure 2.7c do not have holes. Holes in fault surfaces like those shown in Figure 2.7b make it difficult to determine which samples of the seismic image should be used when computing the amplitude differences required for dynamic warping. Holes also make it difficult for the dynamic warping method to enforce the constraints that fault throws vary smoothly. For these reasons, fault throws estimated
using fault surfaces like those shown in Figure 2.7c are more accurate than those for fault surfaces with holes, like those shown in Figure 2.7b.

Figure 2.9: Fault surfaces and fault throws (a) for a 3D seismic image before (b) and after (c) unfaulting. In all image slices, reflectors are more continuous after unfaulting.

Fault surfaces in Figure 2.9a are the same surfaces shown in Figure 2.6c, but colored by fault throws estimated using the method discussed above. We observe that fault throws estimated for each surface vary smoothly, as expected. Also, fault throws for fault F-C are negative because this fault is a reverse fault. Estimated fault throws for faults F-A, F-B, and F-C generally increase in magnitude with depth, while throws for the smaller fault F-D first increase, then decrease with depth. An unfaulting processing described below verifies the accuracy of these estimated fault throws.

With fault throw estimated for each fault sample in a fault surface, we can use the links and the dip vectors $\mathbf{u}$ to walk upward or downward, to determine fault heave for that sample. Fault heave is the horizontal component of a slip vector, and is decomposed into horizontal inline and crossline components. In this way, a slip vector for each fault sample is computed and represented by a vertical component in the traveltime or depth direction, and two horizontal components in inline and crossline directions. The two horizontal components are computed but not shown in this paper.

If the computed fault dip slip vectors are accurate, then we should be able to undo faulting apparent in the seismic image, that is, to correctly align seismic reflectors on opposite sides
of faults.

2.5.2 Unfaulted images

Hale (2013b) uses seismic image unfaulting (Luo and Hale, 2013) to verify the accuracy of estimated dip slips. In their unfaulting method, the dip slips are assumed to be displacements of image samples adjacent to hanging wall sides of faults, and slips for image samples adjacent to footwall sides of faults are assumed to be zero. Because each sample on a fault always lies between two samples of a seismic image, it is easy to locate and set the slips for each pair of footwall and hanging wall samples. Luo and Hale (2013) then simply interpolate all three components of slip vectors for all samples of the seismic image between faults, so that slips away from faults are smoothly varying. Unfortunately, where faults intersect, the assumption that slips on the footwall sides of faults are zero yields unnecessary distortions in the unfaulted seismic image.

Here we use a different method described by Wu et al. (2016) to unfault a seismic image, and thereby verify our estimated dip slips. In this method, vector shifts are computed for all samples in the seismic image by solving partial differential equations derived from the fault slip vectors estimated at faults. This method moves both footwalls and hanging walls, and even faults themselves, simultaneously, to undo faulting with minimal distortion.

Figure 2.9b shows the seismic image with faults colored by estimated fault throws, the vertical components of estimated dip slip vectors. These fault throws, as well as the two horizontal components of the slips, are used in Wu et al. (2016) to obtain the unfaulted image shown in Figure 2.9c. In the unfaulted image, seismic reflectors are well aligned across faults, including the intersecting normal faults and the reverse fault. This unfaulted image illustrates that estimated fault slip vectors are accurate to within the resolution of the seismic image.
Figure 2.10: Fault samples colored by fault likelihood (a) are computed, and linked to form fault surfaces (b).
Figure 2.11: Fault surfaces and fault throws for a 3D seismic image before (a) and after (b) unfaulting. In all image slices, reflectors are more continuous after unfaulting. The red arrows point to a large-slip fault before and after unfaulting.
2.6 A real image example

The synthetic 3D seismic image shown in Figures 2.3–2.9 illustrates our 3D seismic image processing for (1) computing fault images, (2) constructing fault samples and (3) fault surfaces, (4) estimating fault dip slips, and (5) unfaulting the seismic image to assess the accuracy of those slip vectors. This synthetic example also demonstrates that this image processing works for both normal and reverse faults, and for intersecting faults.

A subset of a real 3D seismic image, provided by Kees Rutten and Bob Howard via TNO, is used here as a further demonstration of the same processing. In this real seismic image shown in Figure 2.10 (and in the smaller subset shown in Figure 3.1), many faults are apparent and many of them intersect with others. With such faults, this image is a good example and summary of the methods discussed above.

(1) From this 3D seismic image, images of fault likelihood, strike and dip are first computed by scanning over a range of possible strikes and dips with a simblance-based filter that highlights locally planar discontinuities.

(2) These three fault images are then represented by fault samples, which are displayed as squares oriented by strikes and dips, and colored by fault likelihood in the right-upper panel of Figure 2.10a. Remember that each fault sample corresponds to a seismic image sample in the sampling grid of the seismic image; therefore, the same fault samples can be displayed as a fault likelihood image overlayed with the seismic image slices shown in Figure 2.10a.

(3) The oriented fault samples are then linked to form fault surfaces, displayed in the right-upper panel of Figure 2.10b. Many of these fault surfaces intersect each other, and the differences in strikes for these intersecting faults are approximately 60 degrees. These fault surfaces are really just linked lists of fault samples located within the sampling grid of the seismic image; they appear as surfaces only because the squares representing the fault samples are displayed with sizes large enough to overlap with each other. These linked fault samples can also be displayed as a fault likelihood image overlayed with the seismic image in Figure 2.10b. In the constant-time slice we observe complicated intersections among the
extracted fault surfaces.

Compared to the three slices in Figure 2.10a, some fault samples are removed when constructing surfaces, because they cannot be linked to form surfaces with significant sizes. In this example, we discarded fault surfaces with fewer than 2000 samples. Also, new fault samples are created to fill holes that occur where faults intersect.

(4) These fault surfaces of linked fault samples are further used to estimate fault dip slips. Fault throws (vertical component of slips) are displayed on fault surfaces in the upper-right panel of Figure 3.9. After estimating fault slips, the number of fault surfaces is reduced, because we keep only fault surfaces for which dip slips are significant. Again, each fault sample in a fault surface corresponds to exactly one sample of the 3D seismic image, so fault throws can be displayed as a 3D image overlayed with the seismic image as in Figure 3.9a.

(5) Using the estimated fault dip slip vectors, the seismic image can be unfaulted as shown in Figure 3.9b. In the unfaulted image, seismic reflectors in all image slices are more continuous than those in the original image slices shown in Figure 3.9a. For the fault with large slips highlighted by the red arrow in Figure 3.9a, footwall and hanging wall sides are moved significantly to align the reflectors on these opposite sides, as shown in Figure 3.9b.

2.7 Conclusion

We propose to represent fault surfaces by linked lists of fault samples, each of which corresponds to one and only one seismic image sample. These fault samples can be displayed as 3D fault images (with mostly null values) because they are located on the grid points of the seismic image. Therefore, the processing for faults discussed in this paper is mostly just image processing.

Linked fault samples can also be displayed as fault surfaces by simply increasing sizes of squares used to represent fault samples. These fault surfaces, however, are not triangle or quad meshes, which are unnecessarily complicated for our processing.

Using this simple linked data structure, we construct fault surfaces by simply linking each fault sample and its above, below, left, and right neighbors. These neighbors must
have fault likelihoods, strikes and dips similar to those of the sample for which we search for neighbors. For fault samples with missing neighbors, we propose a method to try to create these neighbors, so as to construct more complete fault surfaces without holes, even when faults intersect. Using complete fault surfaces without holes, fault dip slip vectors can be accurately estimated, and verified by unfaulting the seismic image.

2.8 Acknowledgments

This research is supported by the sponsors of the Consortium Project on Seismic Inverse Methods for Complex Structures. The real 3D seismic image used in this paper was graciously provided by Kees Rutten and Bob Howard, via TNO (Netherlands Organisation for Applied Scientific Research).
CHAPTER 3
MOVING FAULTS WHILE UNFAULTING 3D SEISMIC IMAGES

Modified from a paper published on *Geophysics*  
Xinming Wu\(^1\), Simon Luo\(^2\) and Dave Hale\(^1\)

3.1 Summary

Unfaulting seismic images to correlate seismic reflectors across faults is helpful in seismic interpretation, and is useful for seismic horizon extraction. Methods for unfaulting typically assume that fault geometries need not change during unfaulting. However, for seismic images containing multiple faults and, especially, intersecting faults, this assumption often results in unnecessary distortions in unfaulted images. We developed two methods to compute vector shifts that simultaneously move fault blocks and the faults themselves to obtain an unfaulted image with minimal distortions. For both methods, we use estimated fault positions and slip vectors to construct unfaulting equations for image samples alongside faults, and we construct simple partial differential equations for samples away from faults. We solve these two different kinds of equations simultaneously to compute unfaulting vector shifts that are continuous everywhere except at faults. We test both methods on a synthetic seismic image containing normal, reverse, and intersecting faults, and we also apply one of the methods to a real 3D seismic image complicated by numerous intersecting faults.

3.2 Introduction

It is desirable to undo faulting in a seismic image to align seismic reflectors across faults. For example, from an unfaulted image with more continuous seismic reflectors, seismic horizons can be more easily interpreted. Automatic unfaulting of a seismic image often includes two steps. The first step is to estimate fault slip vectors for faults that are manually or

---

\(^1\)Center for Wave Phenomena, Department of Geophysics, Colorado School of Mines  
\(^2\)BP America Inc., Houston, Texas, USA
automatically extracted from the seismic image. The second step is to extend estimated slip vectors away from samples on faults to all samples in the image, and then simultaneously move fault blocks and even faults to obtain an unfaulted image.

For the first step, several methods have been proposed to estimate fault slip vectors that correlate seismic reflectors on opposite sides of precomputed faults. Fault slip estimated in this way is often dip slip, which is a vector, in the fault dip direction, representing displacement of the hanging wall side of a fault surface relative to the footwall side. In a seismic image, fault strike slip is typically less apparent than dip slip, and is therefore more difficult to estimate by correlating seismic reflectors. To correlate seismic reflectors on the opposite sides of a fault, Aurnhammer and Tonnies (2005) and Liang et al. (2010) propose windowed crosscorrelation methods; Hale (2013b) uses a dynamic warping method that obviates correlation windows.

![Figure 3.1](image.png)

Figure 3.1: A 3D synthetic seismic image with faults colored by fault throws (a), is significantly distorted when unfaulted (b) by moving only fault blocks while fixing fault positions. Faults (especially fault A) must also be moved to obtain an unfaulted image (c) with minimal distortions.

To simplify the second step, Wei et al. (2005) and Wei (2009) assume that fault geometries need not change when unfaulting a seismic image. Luo and Hale (2013) also assume that fault positions are fixed during unfaulting. These assumptions make the unfaulting processing easier, but they might result in unnecessary distortions when unfaulting seismic
images with multiple faults and, especially, intersecting faults. For example, in Figure 3.1, significant distortions are produced in the unfaulted image (Figure 3.1b) by fixing image samples adjacent to faults in the footwalls. Clearly, the faults, and especially fault A, must also be moved to obtain the unfaulted image with less distortion shown in Figure 3.1c.

In this paper, we first use the 3D image processing methods described by Wu and Hale (2016) to automatically compute fault surfaces and dip slip vectors for image samples adjacent to faults. We then introduce two methods to compute unfaulting vector shifts for all samples in a seismic image by solving simple equations derived from the slip vectors. These computed vector shifts simultaneously move footwalls, hanging walls, and even the faults themselves, to undo faulting in a seismic image, with minimal distortion as shown in Figure 3.1c. As an additional test, we apply one of the two methods to a real 3D seismic image complicated by many intersecting faults. The unfaulted image with reflectors that are continuous across faults is then flattened using the unfolding method described by Luo and Hale (2013) to obtain a seismic horizon volume.

3.3 Methods

Prior to unfaulting a 3D seismic image, we must first extract fault surfaces and estimate fault slip vectors. As shown in Figure 3.2, we use the method described by Wu and Hale (2016) to automatically compute fault surfaces (Figure 3.2b) and fault dip slips, the components of fault slips in the fault dip directions (Figure 3.2b). Fault dip slip is a vector, and the vertical component of this vector is fault throw, which is represented by color on the fault surfaces in Figure 3.2b. The horizontal components of slip vectors in inline and crossline directions are not shown in this paper. Fault throw can also be displayed as a 3D image (with mostly null values) overlayed with the seismic image in Figure 3.2c. Note that fault throws are nonnegative for faults A, C and D, but negative for fault B, which indicates that the faults A, C and D are normal faults, while fault B is a reverse fault. For the intersecting faults A and D, the older fault A is dislocated by the younger fault D. Therefore, to undo the faulting for faults A and D, we must move the faults as well as the adjacent fault blocks.
As shown in Figure 3.2c, fault slips are estimated only at the locations of faults. However, to undo faulting apparent in a seismic image without distorting the image, we cannot shift only the image samples adjacent to faults. Instead, we must shift all samples in the image, and move entire fault blocks and even faults themselves. Wei et al. (2005) and Luo and Hale (2013) propose to extend fault slips away from faults, into fault blocks, while fixing fault locations. Without shifting faults, however, these methods cannot correctly undo faulting in an image containing complicated faults, especially intersecting faults, like those shown in Figure 3.2.

We propose two methods to compute vector shifts for all samples in an image, by solving simple equations derived from fault slips on faults, to move faults and fault blocks simultaneously.

![Figure 3.2: Given a 3D seismic image (a), we extract fault surfaces (b) and estimate fault dip slip vectors for each sample on fault surfaces. Faults in (b) and (c) are colored by fault throws, the vertical components of slip vectors.](image)

3.3.1 Mappings between input and unfaulted spaces

Let $f(x)$ denote an input 3D seismic image, a sampled function of coordinates $x \equiv (x_1, x_2, x_3)$ in the input space. To undo faulting in this image, we must find a mapping $x(w)$, where $w \equiv (w_1, w_2, w_3)$ are coordinates in the unfaulted (output) space, and then compute an unfaulted image.
\[ h(w) = f[x(w)]. \]  

(3.1)

We express the mapping \( x(w) \) in terms of a shift vector field \( r(w) \) defined in the unfaulted space:

\[ x(w) = w + r(w). \]  

(3.2)

Therefore, the desired mapping \( x(w) \) can be obtained by solving for the shift vector field \( r(w) \). For any location \( w \) in the sampling grid of the unfaulted space, this mapping \( x(w) \) tells us where to find the corresponding sample in the input space. However, it can be difficult to directly solve for the shift vector field \( r(w) \) in the unfaulted space, because fault locations and slip vectors are computed in the input space.

We assume that the mapping \( x(w) \) from unfaulted coordinates \( w \) to input coordinates \( x \) is reversible. This means that we can find a mapping \( w(x) \) that converts points from the input space to the unfaulted space. We express \( w(x) \) in terms of a shift vector field \( s(x) \) in the input space:

\[ w(x) = x - s(x). \]  

(3.3)

We can usually find this shift vector field \( s(x) \) in the input space by using the fault locations and dips we have in the input space, and thereby obtain the mapping \( w(x) \). However, if applied directly to a uniformly sampled input image \( f(x) \), the mapping \( w(x) \) yields an irregularly sampled unfaulted image \( h(w(x)) = f(x) \). Therefore, we instead use the inverse mapping \( x(w) \) and 3D sinc interpolation of \( f(x) \) to compute a uniformly sampled image \( h(w) = f(x(w)) \).

For these reasons, we first solve for the shift vector field \( s(x) \) in the input space, and then convert \( s(x) \) to the shift vector field \( r(w) \) in the unfaulted space, which is then used to compute the mapping \( x(w) = w + r(w) \) and the unfaulted image \( h(w) \).

Assuming that the mapping between the input and unfaulted spaces is reversible, equations 3.2 and 3.3 imply the following relationship between the shift vector fields \( s(x) \) and
We solve this equation for $r(w)$ using an iterative method. We begin with an initial shift vector field $r_0(w) = s(w)$, and then iteratively update the initial shift vector field to compute $r(w)$:

$$
egin{align*}
  r_0(w) &= s(w) \\
  x_0(w) &= w + r_0(w) \\
  r_1(w) &= s(x_0(w)) \\
  x_1(w) &= w + r_1(w) \\
  & \vdots \\
  r_i(w) &= s(x_{i-1}(w)) \\
  x_i(w) &= w + r_i(w) \\
  & \vdots \\
  r(w) &\approx r_m(w) = s(w + r_{m-1}(w)).
\end{align*}
$$

In this way, we update the shift vector field $r_i(w)$ until the updates are insignificant in the $m$-th iteration, to obtain the shift vector field $r(w) \approx r_m(w)$ in the unfaulted space. This iterative process is fast, as only a nearest neighbor interpolation method is needed when computing $r_i(w) = s(w + r_{i-1}(w))$. In practice, we find that $m = 20$ iterations are sufficient. Therefore, we can efficiently compute $r(w)$ in the unfaulted space, if we already know $s(x)$ in the input space.

To compute the shift vector field $s(x)$ in the input space, we propose two methods that solve simple equations derived from slip vectors estimated at faults.

### 3.3.2 Vector shifts in input space

As discussed by Rice (1983), faults can be considered as surfaces of slip (displacement) discontinuity in surroundings with continuous slip. This means that, when a fault is formed, the slip vector field generating this fault should be continuous in neighboring fault blocks but is discontinuous at the fault. Therefore, to undo faulting apparent in a seismic image, we must compute unfaulting shifts that are also continuous in fault blocks and discontinuous at
faults. Accordingly, we define equations of unfaulting differently for image samples alongside faults and for those elsewhere within fault blocks.

Figure 3.3: A fault slip vector \( t(x_a) \), estimated at each footwall sample adjacent to a fault, tells us how to correlate the image sample at \( x_a \) in the footwall to the corresponding sample \( x_b \) in the hanging wall.

After estimating fault slips shown in Figures 3.2b and 3.2c, we are able to compute unfaulting shifts for the samples adjacent to faults. Figure 3.3 shows an example of a slip vector \( t(x_a) \) estimated at a sample \( x_a \) adjacent to a fault from footwall; this slip vector indicates how to correlate the image sample at \( x_a \) in the footwall to the corresponding sample at \( x_b = x_a + t(x_a) \) in the hanging wall. Image samples \( x_a \) and \( x_b \) must be located at the same position in the unfaulted space:

\[
   w(x_a) = w(x_b),
\]

which can be rewritten using equation 3.3 as

\[
   x_a - s(x_a) = x_b - s(x_b).
\]

Because \( x_b = x_a + t(x_a) \), we have

\[
   s(x_b) - s(x_a) = t(x_a).
\]

Because both the shifts \( s \) and slips \( t \) are vectors, equation 3.8 represents three equations, one for each component, and we can write the three equations as

\[
   s_k(x_b) - s_k(x_a) = t_k(x_a),
\]
where \( k = 1, 2, 3 \) are indices representing the components of vectors in the crossline, inline and vertical directions, respectively.

Recall that we estimate slip vectors everywhere within faults, which means that we have unfaulting equation 3.9 for all image samples alongside faults. Assuming that slip vectors are estimated for \( L \) samples on faults, then we have \( L \) unfaulting equations for each component of our desired vector shifts.

Equation 3.9 applies only to those samples alongside faults. For other samples away from faults, we expect unfaulting shifts to vary slowly and continuously. Thus, derivatives of each component of the vector shift \( s(x) \) should be nearly zero:

\[
\omega(x) \nabla s_k(x) \approx 0, \tag{3.10}
\]

where \( \nabla \) represents the gradient operator, and \( s_k(x)(k = 1, 2, 3) \) represent the three components of vector shifts for all samples in an image. Here, \( \omega(x) \) is a weighting function that is zero at image samples adjacent to faults, and is one elsewhere. Therefore, equation 3.10 is used for all image samples except those adjacent to faults.

Having defined unfaulting equation 3.9 for image samples alongside faults, and the smoothing equation 3.10 for samples elsewhere, we can now solve for the unfaulting shifts \( s(x) \). We propose two methods to simultaneously solve these unfaulting and smoothing equations for \( s(x) \) in two different ways. Both methods work well for the examples in this paper, but they are derived based on different assumptions about the estimated slip vectors and they use the unfaulting equation 3.9 in different ways. Method I assumes that slip vectors are estimated for most samples on faults, but that the estimated slips might be inaccurate for some samples. Method II assumes that slip vectors are picked manually for a limited number of samples on faults, and that these slip vectors are accurate.

3.3.3 Method I

In practice, automatically estimated slip vectors might be inaccurate for some samples on faults. In such a situation, we want to rewrite equation 3.9 as an approximation:
\[ s_k(x_b) - s_k(x_a) \approx t_k(x_a). \] (3.11)

In addition, if we have a measure \( c(x) \) of the quality of the estimated slip vectors at faults, we can use this measure to weight equation 3.11 so that samples with well estimated slips are weighted more than those with poorly estimated slips:

\[ c(x_a)(s_k(x_b) - s_k(x_a)) \approx c(x_a)t_k(x_a). \] (3.12)

For the examples in this paper, the measure \( c(x) \) is fault likelihood (Wu and Hale, 2016), which we compute for every image sample location \( x \) where the slip vector \( t(x) \) is also estimated.

To compute unfaulting shifts for all samples in an image, we solve equations 3.10 and 3.12 simultaneously:

\[ \omega(x) \nabla s_k(x) \approx 0 \]
\[ \beta c(x_a)(s_k(x_b) - s_k(x_a)) \approx \beta c(x_a)t_k(x_a), \] (3.13)

where we have introduced the parameter \( \beta \) to balance the two equations. For all examples in this paper, we use \( \beta = \frac{N}{L} \), where \( L \) is the number of samples on faults and \( N \) is the number of all samples in a seismic image. Although we solve the two equations simultaneously, the second equation is defined only for samples adjacent to faults, where the first equation is disabled because \( \omega(x) \) is zero for those samples.

Because equations 3.13 for the different components \( (k = 1, 2, 3) \) of vector shifts are not coupled with each other, we can solve for each component independently. We use the vertical component \( (k = 3) \) to explain how to solve these equations:

\[ \omega(x) \nabla s_3(x) \approx 0 \]
\[ \beta c(x_a)(s_3(x_b) - s_3(x_a)) \approx \beta c(x_a)t_3(x_a). \] (3.14)

These equations can be represented in matrix-vector form as

\[
\begin{bmatrix}
WG \\
CM
\end{bmatrix} \mathbf{s} \approx \begin{bmatrix}
0 \\
Ct
\end{bmatrix},
\] (3.15)

where \( \mathbf{s} \) is a \( N \times 1 \) vector representing the unknown vertical shifts for a 3D image with \( N \) samples; \( \mathbf{G} \) is a \( 3N \times N \) matrix representing finite-difference approximations of the gradient.
operator; $\mathbf{W}$ is a $3N \times 3N$ diagonal matrix with zeros and ones on the diagonal entries, the zeros corresponding to samples adjacent to faults, and the ones corresponding to samples away from faults; $\mathbf{t}$ is an $L \times 1$ vector containing the vertical component of slip vectors estimated for $L$ ($L < N$) samples on faults; $\mathbf{C}$ is an $L \times L$ diagonal matrix with fault likelihoods scaled by $\beta$ on the diagonal; and $\mathbf{M}$ is an $L \times N$ sparse matrix with mostly zeros, ones for the samples adjacent to faults in hanging walls, and negative ones for the samples adjacent to faults in footwalls.

In total, we have $3N + L$ equations for only $N$ unknowns. Therefore, we might compute a least-squares solution of equation 3.15 by solving the normal equations.
\[ G^\top W^\top W G s + M^\top C^\top C M s = M^\top C^\top C t, \]

(3.16)

where the first term corresponds to the smoothing equation 3.10. In practice, however, fault
dip slips typically vary mainly in dip directions, which are often more consistent in directions
normal to seismic reflectors than in directions parallel to those reflectors. Therefore, instead
of the isotropic smoothing used in equation 3.16, we should smooth less for unfaulting shifts
in directions normal to reflectors than in directions parallel to reflectors.

To implement this anisotropic smoothing of unfaulting shifts, we simply modify the first
term in equation 3.16 by adding a matrix \( D \):

\[ G^\top W^\top D W G s + M^\top C^\top C M s = M^\top C^\top C t. \]

(3.17)

The matrix \( D \) contains spatially varying tensors derived from structure tensors (Fehmers
and Höcker, 2003; Van Vliet and Verbeek, 1995) computed for all image samples. Each
tensor \( T \) represented in the matrix \( D \) is a \( 3 \times 3 \) symmetric positive-definite matrix with
eigen-decomposition

\[ T = \lambda_1 v_1 v_1^\top + \lambda_2 v_2 v_2^\top + \lambda_3 v_3 v_3^\top, \]

(3.18)

where \( v_1 \) is an eigenvector normal to seismic reflectors, \( v_2 \) and \( v_3 \) are eigenvectors that lie
within a plane tangent to seismic reflectors. Eigenvalues \( \lambda_1, \lambda_2, \) and \( \lambda_3 \), all in the range \([0, 1]\),
correspond to eigenvectors \( v_1, v_2, \) and \( v_3, \) respectively. For the examples in this paper, we
set \( \lambda_1 = 0.01, \lambda_2 = \lambda_3 = 1.0 \) to construct the tensor matrix \( D \), so that unfaulting shifts are
smoothed in directions normal to reflectors less than in directions parallel to reflectors.

Note that to solve equation 3.17 for the vertical shifts \( s \), we do not explicitly form
the matrices in this equation. Instead we solve the equation using a conjugate gradient
(CG) method, which requires only the computation of matrix-vector products for matrices
\( G^\top W^\top D W G, M^\top C^\top C M, \) and \( M^\top C^\top C \). Similarly, we can also solve for the horizontal
components of the unfaulting vector shifts in inline and crossline directions.
For example, we use slip vectors, estimated on the fault surfaces shown in Figure 3.2, to construct the coefficients in equation 3.17. Then solving this equation, we compute the vertical, inline, and crossline components of the unfaulting shifts shown in Figures 3.4a, 3.4b and 3.4c, respectively. We observe that the shifts are discontinuous at faults and continuous elsewhere, as expected.

### 3.3.4 Method II

For method I, we assumed that fault slip vectors are estimated using an automatic method for most samples on faults, and, as a result, might be inaccurate for some samples. However, in an interactive interpretation system, one might manually pick pairs of points, for example $x_a$ and $x_b$ in Figure 3.3, alongside a fault, and then simply compute corresponding slip vectors $t(x_a) = x_b - x_a$.

In this case, we expect the unfaulting equation 3.9 with interpreted slip vectors to be strictly satisfied for manually picked pairs of points alongside a fault. At the same time, however, we still expect shifts to vary smoothly within fault blocks, for all image samples located away from faults. Therefore, for method II, instead of solving equation 3.17, we compute the unfaulting shifts by solving

$$G^T W^T D W G s = 0 \quad \text{subject to} \quad M s = t.$$  \hspace{1cm} (3.19)

As discussed by Wu and Hale (2015b), we use a preconditioned CG method to solve this linear system with hard constraints. The unfaulting equation $M s = t$ is implemented with simple preconditioners in the CG method; the details of constructing such preconditioners are discussed by Wu and Hale (2015b). Starting with initial shifts that satisfy the unfaulting equation $M s = t$, the CG iterates update the shifts for all samples, while the preconditioners guarantee that the updated shifts always satisfy the unfaulting equation after each iteration.

To test this method, we used our automatically estimated fault slips to construct the unfaulting equation $M s = t$ for all samples alongside faults, and compute initial shifts with which the CG method begins. The computed vertical, inline and crossline components of
vector shifts are shown in Figures 3.5a, 3.5b and 3.5c, respectively. Similar to the shifts computed using method I, each component of shifts computed using this method is discontinuous at faults and smoothly varying elsewhere.

Figure 3.6: Vertical (a), inline (b) and crossline (c) components of unfaulting shifts in the unfaulted space are converted from those in the input space shown in Figure 3.4. The discontinuities on each component of shifts are displaced relative to those in Figure 3.4.

Figure 3.7: Vertical (a), inline (b) and crossline (c) components of unfaulting shifts in the unfaulted space are converted from those in the input space shown in Figure 3.5. The discontinuities on each component of shifts are displaced relative to those in Figure 3.5.

3.3.5 Vector shifts in the unfaulted space

The shifts $s(x)$ computed by the two methods above are all in the input space. We must map them into the unfaulted space before unfaulting the seismic image. We obtain the corresponding vector shifts $r(w)$ in the unfaulted space using the efficient iteration method in equation 3.5.
Figure 3.8: The input synthetic seismic image (a) is unfaulted (b) using shifts in Figure 3.6 computed by method I, and (c) using shifts in Figure 3.7 computed by method II.

Figure 3.6 shows all components of vector shifts $\mathbf{r}(\mathbf{w})$ obtained in this way from the vector shifts $\mathbf{s}(\mathbf{x})$ (Figure 3.4) computed in the input space using method I. Figure 3.7 shows all components of vector shifts $\mathbf{r}(\mathbf{w})$ converted from the vector shifts $\mathbf{s}(\mathbf{x})$ (Figure 3.5) computed in the input space using method II. Before conversion, we observe that discontinuities in each component of shifts coincide with faults in the input space, as in Figures 3.4 and 3.5. However, after converting shifts to the unfaulted space, the discontinuities on each component of shifts in Figures 3.6 and 3.7 are displaced relative to those in Figures 3.4 and 3.5.

Using the converted vector shifts $\mathbf{r}(\mathbf{w})$ in Figures 3.6 (method I) and 3.7 (method II), we obtain the corresponding unfaulting mapping $\mathbf{x}(\mathbf{w}) = \mathbf{w} + \mathbf{r}(\mathbf{w})$ and then compute the unfaulted images shown in Figures 3.8b (method I) and 3.8c (method II). In both unfaulted images, seismic reflectors are more continuous than those in the input seismic image (Figure 3.8a). We also observe that the faults are shifted in the unfaulted space, relative to the input space. For example, fault A is dislocated in the original seismic image (Figure 3.8a) by its intersecting fault, but is relocated in both unfaulted images (Figures 3.8b and 3.8c) computed using two different methods.

As shown in Figures 3.8b and 3.8c, both methods provide unfaulted images with minimal distortions, because slip vectors (Figure 3.2) are estimated accurately for all faults in this
synthetic example. In practice, however, we suggest using method I when slip vectors are estimated using an automatic method for numerous samples on faults, because such slip vectors might be inaccurate for some samples. Large errors in slip vectors will yield large errors in the unfaulting shifts computed using method II, because the unfaulting equations with slip vectors serve as hard constraints for this method. For slip vectors with errors, method I is preferred, because it computes a least-squares solution of the unfaulting equations, which can be weighted according to some measure of the quality of estimated slips.

If instead fault slip vectors are manually interpreted for only a limited number of samples alongside faults, then we suggest method II. For this method, the unfaulting equations constructed from the interpreted slip vectors serve as hard constraints for computing unfaulting shifts; therefore, the resulting unfaulted image is guaranteed to be consistent with the interpretation.

### 3.4 Application

The synthetic examples shown in Figure 3.8 demonstrate that both methods work well in unfaulting normal, reverse, and intersecting faults. As an additional test, method I was further applied to a real seismic image complicated by intersecting faults.

From the 3D seismic image shown in Figure 3.9a, we first used the methods described by Wu and Hale (2016) to compute fault surfaces and dip slip vectors. Fault throws, the vertical components of dip slips, are displayed in color in Figure 3.9a. Note that fault throws are nonnegative, which indicates that the faults shown here are normal faults. We observe that most fault surfaces intersect others, and from the horizontal slice in Figure 3.9a, the strike angles for the intersecting faults differ by approximately 60 degrees.

Using the computed fault surfaces and slip vectors, we then computed unfaulting vector shifts $r(w)$ in unfaulted space using method I. The vertical components of the shifts are displayed in Figure 3.9b. The inline and crossline components of the shifts are not shown. The intersections of faults are apparent in the horizontal slice of the vertical shifts shown in Figure 3.9b.
Figure 3.9: Fault surfaces and slip vectors (a) are first estimated from a 3D seismic image, and then are used to compute unfaulting vector shifts (b) used below in image unfaulting. Only vertical components of vectors are shown here.
Figure 3.10: A 3D seismic image before (a) and after (b) unfaulting. In all image slices, seismic reflectors are more continuous after unfaulting. For the large-throw fault highlighted by a red arrow in (a), the corresponding fault blocks are significantly moved in (b) to align seismic reflectors on opposite sides of this fault.
Figure 3.11: Composite shifts (a) are computed and then used to obtain an unfaulted and unfolded image (b).
Figure 3.12: Two horizon surfaces (colored by depth) are extracted using composite shift vectors that map an image from input space to unfaulted and unfolded space. The vertical component of the composite shift vectors is displayed in Figure 3.11a.

Using the unfaulting vector shifts \( r(w) \), we then compute the unfaulting mapping \( x(w) = w + r(w) \), which undoes the faulting in the seismic image (Figure 3.10a) to produce the unfaulted image shown in Figure 3.10b. In this unfaulted image, seismic reflectors in all image slices are more continuous across faults than those in the original image slices shown in Figure 3.10a. For the fault with large slips highlighted by the red arrow in Figure 3.10a, footwall and hanging wall sides are moved significantly to align the reflectors on opposite sides of the fault, as shown in Figure 3.10b.

For an unfaulted image with seismic reflectors that are continuous across faults, seismic horizon interpretation is more straightforward, for either manual or automatic methods. Here we used the method described by Luo and Hale (2013) to compute vector shifts that undo the folding in the unfaulted image (Figure 3.10b), to obtain the unfolded image shown in Figure 3.11b. In the slices of the unfolded image shown in Figure 3.11b, seismic reflectors are horizontal. As discussed by Luo and Hale (2013), using the unfolding vector shifts together with unfaulting vector shifts, we can compute composite vector shifts, which enable us to directly map the input seismic image to the unfaulted and unfolded space. The vertical
components of the computed composite vector shifts are displayed in Figure 3.11a.

Using the composite vector shifts, we are able to extract any number of seismic horizons from the input seismic image in the input space, as discussed by Luo and Hale (2013). Figure 3.12 shows two seismic horizons extracted using the computed vector shifts. Our unfaulting processing facilitates the extraction of such complicated horizon surfaces by aligning seismic reflector across faults.

3.5 Conclusion

We have described two methods to automatically undo faulting in 3D seismic images. Both methods require precomputed fault positions and slip vectors at faults. Both methods efficiently compute vector shifts that simultaneously move fault blocks and faults themselves to undo faulting in seismic images. We suggest using method I when fault slips are estimated automatically for most samples at faults, because this method computes a least-squares solution of the unfaulting equations constructed from estimated slips. Method II is preferable if fault slips are manually interpreted for only a limited number of samples at faults, because this method considers the interpreted slips as hard constraints when computing unfaulting shifts.

One limitation of both methods is that they do not truly reverse the geologic deformation of faulting. We construct simple partial differential equations for samples away from faults in fault blocks to obtain smooth unfaulting shifts for these samples. The unfaulting shifts are allowed to vary more significantly in directions normal to seismic reflectors than in directions parallel to reflectors by using spatially variant tensor fields as coefficients in these partial differential equations. Although these simple equations can be solved efficiently and unfaulted images appear reasonable, it might be possible and preferable to use a more geologically and geomechanically correct way to compute unfaulting shifts for samples away from faults.
3.6 Acknowledgments

This research is supported by the sponsors of the Consortium Project on Seismic Inverse Methods for Complex Structures. The real 3D seismic image used in this paper was graciously provided by Kees Rutten and Bob Howard, via TNO (Netherlands Organisation for Applied Scientific Research).
CHAPTER 4
3D SEISMIC IMAGE PROCESSING FOR UNCONFORMITIES

Modified from a paper published in Geophysics, 2015, 80 (2), IM35-IM44
Xinming Wu\(^1\) and Dave Hale\(^1\)

4.1 Summary

In seismic images, an unconformity can be first identified by reflector terminations (i.e., truncation, toplap, onlap or downlap) and then be traced downdip to its corresponding correlative conformity, or updip to a parallel unconformity, for example in topsets. Unconformity detection is a significant aspect of seismic stratigraphic interpretation, but most automatic methods work only in 2D and can only detect angular unconformities with reflector terminations. Moreover, unconformities pose challenges for automatic techniques used in seismic interpretation. First, it is difficult to accurately estimate normal vectors or slopes of seismic reflectors at an unconformity with multi-oriented structures due to reflector terminations. Second, seismic flattening methods cannot correctly flatten reflectors at unconformities that represent hiatuses or geologic age gaps. To address these challenges, we first propose a 3D unconformity attribute computed from a seismic amplitude image to detect unconformities by highlighting both angular unconformities and corresponding parallel unconformities or correlative conformities. These detected unconformity surfaces are further used as constraints for a structure-tensor method to more accurately estimate seismic normal vectors at unconformities. Finally, using detected unconformities as constraints and more accurate normal vectors, we can better flatten seismic images with unconformities.

\(^1\)Center for Wave Phenomena, Department of Geophysics, Colorado School of Mines
4.2 Introduction

An unconformity is a non-depositional or erosional surface separating older strata below from younger strata above, and thus represents a significant gap in rock record (Vail et al., 1977). Unconformity extraction from seismic images is important for seismic stratigraphic interpretation, because unconformities represent discontinuities in otherwise continuous deposits and hence serve as boundaries when interpreting seismic sequences that represent successively deposited layers.

In addition, the detected unconformities can be applied as constraints to improve a structure-tensor method (Fehmers and Höcker, 2003; Van Vliet and Verbeek, 1995) for more accurately estimating normal vectors of seismic reflectors at unconformities with multi-oriented structures, and improve a seismic image flattening method (Lomask et al., 2006; Luo and Hale, 2013; Parks, 2010; Wu and Hale, 2015b) to more accurately flatten reflectors at unconformities with geologic age gaps.

4.2.1 Unconformity detection

Seismic coherence (Bahorich and Farmer, 1995), highlights reflector discontinuities, is commonly used to detect faults, channel edges, and other lateral changes in waveform. Although sensitive to unconformities, coherence vertically smears the response over the computation window and unconformities usually appear as vertical changes in waveform. Barnes et al. (2000) and Hoek et al. (2010) propose an unconformity attribute that measures the degree of seismic reflector convergence (or divergence), and thereby highlights the termination areas of an unconformity. Smythe et al. (2004) introduce a SPICE (spectral image of correlative events) attribute to obtain stratigraphic details by highlighting discontinuities in band-limited seismic data. All of these methods process a seismic image locally (Ringdal, 2012) to compute unconformity attributes that can highlight an unconformity within its termination area, but cannot detect its corresponding parallel unconformities or correlative conformities.
Ringdal (2012) proposes a global method that first extracts a 2D flow field that is everywhere tangent to reflectors in a 2D seismic image. Then the flow field is used to compute an unconformity probability image by repeating the following processing for each sample: (1) four seeds are first placed at the four neighbors of the sample in the 2D flow field; (2) the four seeds then move along the flow field to produce trajectories; (3) the separation rate of the trajectories is calculated and (4) this separation is converted to an unconformity probability for that sample. The advantage of this method is that it can detect a parallel unconformity or correlative conformity by using long trajectories that extend from the parallel unconformity or correlative conformity to the corresponding angular unconformity. The disadvantage is that, to detect such a correlative conformity or parallel unconformity, the trajectories are required to start from the parallel area (parallel unconformity highlighted by the blue ellipse in Figure 1a) and end in the non-parallel area (termination highlighted by the green ellipse in Figure 1a). Another disadvantage is the trajectory extraction along a flow field applies only for 2D images. For 3D seismic images, this method processes inline and crossline slices separately throughout the volume to compute an unconformity probability volume.

4.2.2 Seismic normal vector estimation at unconformities

Orientation vector fields, such as vectors normal to or slopes of seismic reflectors, are useful for seismic interpretation. For example, estimated orientation information is used to control slope-based (Fomel, 2002) and structure-oriented (Fehmers and Höcker, 2003; Hale, 2009) filters so that they smooth along reflectors to enhance their coherencies. Seismic normal vectors or slopes are also used to track horizons (de Groot et al., 2010) and to flatten (Lomask et al., 2006; Parks, 2010) or unfold (Luo and Hale, 2013) seismic images, or to generate horizon volumes (Wu and Hale, 2015b).

Structure tensors (Fehmers and Höcker, 2003; Van Vliet and Verbeek, 1995) or plane-wave destruction filter (Fomel, 2002) have been proposed to estimate seismic normal vectors or slopes. These methods can accurately estimate orientation vectors for structures with only one locally dominant orientation. This means that they can correctly estimate the
normal vectors (or slopes) of the reflectors in conformable areas of a seismic image, but for an angular unconformity where two different structures meet, these methods yield smoothed vectors that represent averages of orientations across the unconformity.

4.2.3 Seismic image flattening at unconformities

Seismic image flattening (Lomask et al., 2006; Parks, 2010; Wu and Hale, 2015b) or unfolding (Luo and Hale, 2013) methods are applied to a seismic image to obtain a flattened image in which all seismic reflectors are horizontal. From such a flattened seismic image, all seismic horizons can be identified by simply slicing horizontally.

Extracting horizons terminated by faults or unconformities is generally difficult for these methods. Luo and Hale (2013) extract horizons across faults by first unfaulting a seismic image; Wu and Hale (2015b) do the same by placing control points on opposite sides of faults. However, none of these methods correctly flattens a seismic image with unconformities, which should produce gaps in the flattened image. Wu and Zhong (2012a,b) flatten a seismic image with unconformities by using a relative geologic time (RGT) volume generated with a phase unwrapping method, but all the unconformities in the seismic image have to be manually interpreted to constrain the phase unwrapping.

4.2.4 This paper

In this paper, we first propose a method to automatically detect an unconformity, complete with its termination area and corresponding parallel unconformities or correlative conformities. We then estimate seismic normal vectors at unconformities by using the detected unconformities as constraints. Finally, we flatten seismic images containing unconformities by using these estimated seismic normal vectors and constraints derived from detected unconformities.
4.3 Unconformity detection

In manual 3D seismic stratigraphic interpretation, an unconformity is first recognized as a boundary at which seismic reflectors terminate by truncation, toplap, onlap, or downlap, and then is traced into parallel reflections either downdip to its correlative conformity, or updip into topsets. Therefore, to obtain a complete unconformity, an automatic method should be able to detect both the termination areas (green ellipse in Figure 4.1a) and parallel unconformities (blue ellipse in Figure 4.1a) or correlative conformities corresponding to the unconformity.

Figure 4.1: A 2D synthetic seismic image (a) with an unconformity (red curve) that is manually interpreted from the termination area to its corresponding parallel unconformity. The estimated seismic normal vectors (red segments in (b)) are smoothed within the termination area, and therefore are incorrect, compared to the true seismic normal vectors (cyan segments in (b)) that are discontinuous within that area.

We propose an unconformity attribute that measures differences between two seismic normal vector fields computed from two structure-tensor fields. The first structure tensor field is computed using a vertically causal smoothing filter. The second one is computed using a vertically anti-causal smoothing filter. This attribute can detect an unconformity by highlighting both its termination areas and its corresponding parallel unconformities or correlative conformities.
4.3.1 Structure tensor

The structure tensor (Fehmers and Höcker, 2003; Van Vliet and Verbeek, 1995) can be used to estimate seismic normal vectors that are perpendicular to seismic reflectors. For a 2D image, the structure tensor $T$ for each sample is a $2 \times 2$ symmetric positive-semidefinite matrix constructed as the smoothed outer product of image gradients:

$$T = \langle gg^\top \rangle_{h,v} = \begin{bmatrix} \langle g_1 g_1 \rangle_{h,v} & \langle g_1 g_2 \rangle_{h,v} \\ \langle g_1 g_2 \rangle_{h,v} & \langle g_2 g_2 \rangle_{h,v} \end{bmatrix},$$  

(4.1)

where $g = [g_1 \ g_2]^\top$ represents the image gradient vector computed for each image sample; $\langle \cdot \rangle_{h,v}$ represents smoothing for each outer-product element in both horizontal (subscript $h$) and vertical (subscript $v$) directions. These horizontal and vertical smoothing filters are implemented with 1-D recursive Gaussian filters (Hale, 2006) with corresponding half-widths $\sigma_h$ and $\sigma_v$.

As shown by Fehmers and Höcker (2003), the seismic normal vector for each image sample can be estimated from the eigen-decomposition of the structure tensor $T$:

$$T = \lambda_u uu^\top + \lambda_v vv^\top,$$  

(4.2)

where $u$ and $v$ are unit eigenvectors corresponding to eigenvalues $\lambda_u$ and $\lambda_v$ of $T$.

We choose $\lambda_u \geq \lambda_v$, so that the eigenvector $u$, which corresponds to the largest eigenvalue $\lambda_u$, indicates the direction of highest change in image amplitude, and therefore is perpendicular to locally linear features in an image, while the orthogonal eigenvector $v$ indicates the direction that is parallel to such features. In other words, the eigenvector $u$ is the seismic normal vector that is perpendicular to seismic reflectors in a seismic image, and the eigenvector $v$ is parallel to the reflectors.

4.3.2 Smoothing

The structure tensor $T$ given in equation 4.1 can be used to accurately estimate the local orientation of structures in an image where there is only one locally dominant orientation present. However, for multi-oriented structures such as an unconformity where seismic reflec-
tors terminate (green ellipse in Figure 4.1a), this structure tensor provides a local average of the orientations of structures. The seismic normal vectors (magenta segments in Figure 4.1b) estimated from $T$ are smoothed near the termination area, whereas the true normal vectors (cyan segments in Figure 4.1b) are discontinuous across the unconformity.

At an unconformity where seismic reflectors terminate (green ellipse in Figure 4.1a), structures of reflectors above the unconformity are different from those of reflectors below. Therefore, if we compute structure tensors using vertically causal smoothing filters, which average structures from above, we will obtain normal vectors at the unconformity that are different from those obtained using vertical anti-causal filters, which average structures from below.

With a vertically causal filter, the structure tensor computed for each sample represents structures averaged using only samples above. We define such a structure tensor as

$$\mathbf{T}_c = \begin{bmatrix} \langle g_1 g_1 \rangle_{h,vc} & \langle g_1 g_2 \rangle_{h,vc} \\ \langle g_1 g_2 \rangle_{h,vc} & \langle g_2 g_2 \rangle_{h,vc} \end{bmatrix}, \quad (4.3)$$

where $\langle \cdot \rangle_{h,vc}$ represents horizontal Gaussian (subscript $h$) and vertically causal (subscript $vc$) smoothing filters.

With a vertically anti-causal filter, the structure tensor computed for each sample represents structures averaged using only samples below. We define such a structure tensor as

$$\mathbf{T}_a = \begin{bmatrix} \langle g_1 g_1 \rangle_{h,va} & \langle g_1 g_2 \rangle_{h,va} \\ \langle g_1 g_2 \rangle_{h,va} & \langle g_2 g_2 \rangle_{h,va} \end{bmatrix}, \quad (4.4)$$

where the subscript $va$ denotes a vertically anti-causal smoothing filter.

### 4.3.3 Vertical smoothing

To compute two structure-tensor fields that differ significantly at an unconformity, the causal smoothing filter that averages from above should smooth along the direction perpendicular to the structures above the unconformity, while the anti-causal filter should smooth along the direction perpendicular to the structures below the unconformity. Here, we sim-
ply use vertically causal and anti-causal filters because unconformities tend to be close to horizontal in seismic images. We implement these two filters with one-sided exponential smoothing filters, which are efficient and trivial to implement.

A one-sided causal exponential filter for input and output sequences \(x[i]\) and \(y[i]\) with lengths \(n\) can be implemented in C++ (or Java) as follows:

```cpp
float b = 1.0f-a;
float yi = y[0] = x[0];
for (int i=1; i<n; ++i)
    y[i] = yi = a*yi+b*x[i];
```

Similarly, a one-sided anti-causal exponential filter can be implemented as follows:

```cpp
float b = 1.0f-a;
float yi = y[n-1] = x[n-1];
for (int i=n-2; i>=0; --i)
    y[i] = yi = a*yi+b*x[i];
```

The parameter \(a\) in these two one-sided exponential filters controls the extent of smoothing. In all examples, we use \(a=0.8\), which for low frequencies approximates a half Gaussian filter with half-width \(\sigma = 6\) samples.

From structure-tensor fields \(T_c\) and \(T_a\) computed for the same seismic image using vertically causal and anti-causal smoothing filters, respectively, we estimate two seismic normal vector fields \(u_c\) and \(u_a\). As shown in Figure 4.2a, the two seismic normal vector fields \(u_c\) (green segments in Figure 4.2a) and \(u_a\) (yellow segments in Figure 4.2a) are identical in conformable areas with parallel seismic reflectors, because orientation of structures locally averaged from above (used to compute \(T_c\)) are identical to orientation of structures averaged from below (used to compute \(T_a\)). However, at the termination area of an unconformity, the two vector fields are different because the structure tensors \(T_c\) computed with structures locally averaged from above should be different from \(T_a\) computed with structures locally averaged from below.
Therefore, as shown in Figure 4.2a, the difference between estimated normal vector fields $u_c$ and $u_a$ provides a good indication of the termination area of an unconformity. However, a complete unconformity, that is, a curve (in 2D) or surface (in 3D) with geologic age gaps, extends from its termination area updip or downdip into parallel reflectors. Thus we should extend normal vector differences from the termination area, where these differences originate, into the corresponding parallel unconformity or correlative conformity.

### 4.3.4 Structure-oriented smoothing

To detect a correlative conformity or parallel unconformity, we extend vector differences (between $u_c$ and $u_a$) at an unconformity from its termination area to its correlative conformity or parallel unconformity, by replacing the horizontal Gaussian smoothing filter in equations 4.3 and 4.4 with a structure-oriented smoothing filter (Hale, 2009) when computing structure tensors.

Then, the structure tensors $T_{s,c}$ and $T_{s,a}$, computed with a laterally structure-oriented filter and vertically causal and anti-causal filters, are defined by

\[
T_{s,c} = \begin{bmatrix}
\langle g_1 g_1 \rangle_{s,vc} & \langle g_1 g_2 \rangle_{s,vc} \\
\langle g_1 g_2 \rangle_{s,vc} & \langle g_2 g_2 \rangle_{s,vc}
\end{bmatrix},
\]

and

\[
T_{s,a} = \begin{bmatrix}
\langle g_1 g_1 \rangle_{s,va} & \langle g_1 g_2 \rangle_{s,va} \\
\langle g_1 g_2 \rangle_{s,va} & \langle g_2 g_2 \rangle_{s,va}
\end{bmatrix},
\]

where the subscript $s$ represents a structure-oriented filter that smoothes along reflectors in a seismic image. Note that the structure-oriented smoothing is generally more expensive than the vertically causal and anti-causal smoothing. We therefore first apply the structure-oriented smoothing filter to each element of $gg^\top$ to obtain $T_s = \langle gg^\top \rangle_s$, which then is smoothed separately by vertically causal and anti-causal filters to obtain $T_{s,c}$ and $T_{s,a}$, respectively. By doing this, we apply the relatively expensive structure-oriented smoothing only once. However, if we first apply the vertically causal and anti-causal smoothing to compute the two differently smoothed outer products $\langle gg^\top \rangle_c$ and $\langle gg^\top \rangle_a$, we then need to
apply the structure-oriented smoothing twice to obtain two structure-tensor fields $T_{s,c}$ and $T_{s,a}$.

As discussed by Hale (2009, 2011), to obtain a smoothed output image $q(x)$ from an input $p(x)$, the structure-oriented smoothing method solves a finite-difference approximation to the following partial differential equation:

$$q(x) - \frac{\sigma^2}{2} \nabla \cdot D(x) \cdot \nabla q(x) = p(x),$$  

(4.7)

where $D(x)$ is a diffusion-tensor field that shares the eigenvectors of the structure tensor computed from an image, and therefore orients the smoothing along image structures. Similar to the half-width $\sigma$ in a Gaussian smoothing filter, the parameter $\sigma$ controls the extent of smoothing.

In 2D, we use the eigenvectors $u(x)$ and $v(x)$, estimated using the structure tensors shown in equation 4.1, to construct our diffusion-tensor field

$$D(x) = \lambda_u(x)u(x)u^\top(x) + \lambda_v(x)v(x)v^\top(x).$$  

(4.8)

Then, because eigenvectors $u(x)$ and $v(x)$ are perpendicular and parallel to seismic reflectors, respectively, we can control the structure-oriented filter to smooth along reflectors by setting the corresponding eigenvalues $\lambda_u(x) = 0$ and $\lambda_v(x) = 1$ for all tensors in $D(x)$. In 3D, a structure tensor $T$ for each image sample is a $3 \times 3$ matrix, from which the eigen-decomposition provides 3 eigenvectors $u$, $v$ and $w$, where $u$ is orthogonal to locally planar features, both $v$ and $w$ lie within the planes of any locally planar features. We then construct the structure-oriented diffusion-tensor field in 3D as $D(x) = \lambda_u(x)u(x)u^\top(x) + \lambda_v(x)v(x)v^\top(x) + \lambda_w(x)w(x)w^\top(x)$, and set $\lambda_u(x) = 0$ and $\lambda_v(x) = \lambda_w(x) = 1$ so that the filter diffuses or smoothes image features along structures.

As indicated by the seismic normal vectors shown in Figure 4.1b, normal vectors (magenta segments) estimated using the structure tensors computed in equation 4.1 are inaccurate at unconformities. However, they are accurate in conformable areas, including the area near the parallel unconformity. Thus the structure tensors in equation 4.1 are adequate for
constructing diffusion-tensors $\mathbf{D}(\mathbf{x})$ for structure-oriented smoothing along seismic reflectors, including those near the parallel unconformity and correlative conformity corresponding to an unconformity. As discussed in Hale (2009), using structure-oriented diffusion tensors $\mathbf{D}(\mathbf{x})$ for the smoothing filter in equation 4.7, the smoothing-filter weights are largest along curvilinear trajectories that coincide with image structures, which means that the filter diffuses or extends image values from high to low along structures. Therefore, by applying such a structure-oriented filter to the elements of the structure tensors $\mathbf{T}_{s,c}$ and $\mathbf{T}_{s,a}$, the structure-oriented smoothing-filter weights extend structural differences, originating within the termination area of an unconformity, into the corresponding parallel unconformity and correlative conformity. Because $\sigma$ in equation 4.7 controls the extend of smoothing, to detect parallel unconformities or correlative conformities that extend long away from an angular unconformity, we need use a large $\sigma$. In practice, we use $\sigma = n_x$ for 2D and $\sigma = \max(n_x, n_y)$ for 3D, where $n_x$ and $n_y$ are the number of samples in the horizontal dimensions of an input seismic image.

Figure 4.2: Two different seismic normal vector fields estimated using structure tensors computed with vertically causal (green segments) and anti-causal (yellow segments) smoothing filters. In (a), the vector fields differ only within the termination area of the unconformity; in (b), these vector differences are extended to the parallel unconformity.
As shown in Figure 4.2, using structure tensors $T_c$ and $T_a$ computed with a horizontal Gaussian filter and vertically causal and anti-causal filters, the estimated seismic normal vectors $u_c$ (green segments in Figure 4.2a) and $u_a$ (yellow segments in Figure 4.2a) differ only within the termination area of the unconformity. Using structure tensors $T_{s,c}$ and $T_{s,a}$ computed with a structure-oriented smoothing filter instead of a horizontal Gaussian filter, the differences between the estimated seismic normal vectors $u_{s,c}$ (green segments in Figure 4.2b) and $u_{s,a}$ (yellow segments in Figure 4.2b) are extended from the termination area to the parallel unconformity.

In summary, by first applying a structure-oriented filter to each structure-tensor element of $gg^T$, we extend any structure differences, which originate within the termination area of an unconformity, to its corresponding parallel unconformity and correlative conformity. Then, applying vertically causal and anti-causal filters for each structure-tensor element, we compute two different structure-tensor fields $T_{s,c}$ and $T_{s,a}$ with seismic normal vector fields $u_{s,c}$ and $u_{s,a}$ that differ within both the termination area and the corresponding correlative conformity and parallel unconformity. Finally, the differences between the two estimated vector fields $u_{s,c}$ and $u_{s,a}$ can be used as an unconformity attribute that highlights the angular unconformity and its corresponding parallel unconformities or correlative conformities.

### 4.3.5 Unconformity likelihood

As shown in Figure 4.2b, the vectors $u_{s,c}$ (green segments) and $u_{s,a}$ (yellow segments) are identical everywhere except at the unconformity, including its termination area and parallel unconformity. Therefore, we define an *unconformity likelihood* attribute $g$, that evaluates the differences between $u_{s,c}$ and $u_{s,a}$, to highlight unconformities:

$$g \equiv 1 - (u_{s,c} \cdot u_{s,a})^p.$$  \hspace{1cm} (4.9)

The power $p$ ($p > 1$) is used to increase the contrast between samples with low and high unconformity likelihoods.
Figure 4.3: Unconformity likelihoods, an attribute that evaluates differences between two estimated seismic normal vector fields (yellow and green segments in Figure 4.2b), before (a) and after (b) thinning highlight both the termination area and parallel unconformity.

Figure 4.4: Unconformity likelihoods before (a) and after (b) thinning.

Using a process similar to that used by Hale (2013b) for extracting ridges of fault likelihoods, we extract ridges of unconformity likelihood by simply scanning each vertical column of the unconformity likelihood image (Figure 4.3a), preserving only local maxima, and setting unconformity likelihoods elsewhere to zero. Figure 4.3b shows that ridges of unconformity likelihood coincide with the unconformity that appears in the synthetic seismic image.

For a 3D seismic image, following the same process as above, we compute an unconformity-likelihood volume as shown in Figure 4.5, which correctly highlights two apparent unconformities. In the time slices of unconformity likelihoods before and after thinning, we observe that samples in the lower-left and upper-right areas, separated by high unconformity like-
Figure 4.5: Unconformity likelihoods before (a) and after (b) thinning. Thinned unconformity likelihoods form unconformity surfaces as shown in the top-right panel in (b).
lihoods, suggest different seismic facies. This indicates that they belong to two different depositional sequences that have different geologic ages.

From ridges of unconformity likelihoods (Figure 4.5b), we connect adjacent samples with high unconformity likelihoods to form unconformity surfaces as shown in upper-right panel of Figure 4.5b.

4.4 Applications

We first use unconformity likelihoods as constraints to more accurately estimate seismic normal vectors at unconformities. Then, by using more accurate normal vectors and unconformity likelihoods as constraints in our seismic image flattening method, we are able to better flatten an image containing unconformities.

4.4.1 Estimation of seismic normal vectors at unconformities

Using structure tensors computed with horizontal and vertical Gaussian filters as shown in equation 4.1, we find smoothed seismic normal vectors (magenta segments in Figure 4.1b) in the termination area, because discontinuous structures across the unconformity are smoothed by symmetric Gaussian filters. Therefore, to obtain correct normal vectors (cyan segments in Figure 4.1b) that are discontinuous in the termination area, we must use more appropriate filters to compute structure tensors.

To preserve structure discontinuities, we compute the structure tensors using horizontal and vertical filters that do not smooth across unconformities:

\[ T = \begin{bmatrix} \langle g_1 g_1 \rangle_{sh,sv} & \langle g_1 g_2 \rangle_{sh,sv} \\ \langle g_1 g_2 \rangle_{sh,sv} & \langle g_2 g_2 \rangle_{sh,sv} \end{bmatrix}, \quad (4.10) \]

where the \( \langle \cdot \rangle_{sh,sv} \) represent horizontal (subscript \( sh \)) and vertical (subscript \( sv \)) filters that vary spatially, and for which the extent of smoothing is controlled by the thinned unconformity likelihoods.

The horizontal and vertical filters are similar to the edge-preserving smoothing filter discussed in Hale (2011):
\[ q(x) - \frac{\sigma^2}{2} \nabla \cdot c^2(x) \cdot \nabla q(x) = p(x). \] (4.11)

We compute \( c(x) = 1 - g_t(x) \) to prevent this filter from smoothing across unconformities. \( g_t(x) \) is a thinned unconformity likelihood image as shown in Figure 4.4b, which has large values (close to 1) only at unconformities, and zeros elsewhere.

Figure 4.6: Vertical \((u_1)\) and horizontal \((u_2)\) components of the true \((a, d)\) normal vectors of the synthetic image \((\text{Figure 4.1a})\), the estimated normal vectors \((c, f)\) with the detected unconformity \((\text{Figure 4.3b})\) as constraints are more accurate than those \((b, e)\) without constraints.

Figures 4.6a and 4.6d show vertical and horizontal components of the true normal vectors for the synthetic image shown in Figure 4.1a. We observe that both the two components are discontinuous at the unconformity. However, using the conventional structure-tensor method as in equation 4.1, the two components \((\text{Figures 4.6b and 4.6e})\) of the estimated seismic normal vectors are smooth and therefore are inaccurate at the unconformity. Using the improved structure-tensor method constrained by the unconformity likelihoods \((\text{Figure 4.3})\) as in equation 4.10, the estimated seismic normal vectors shown in Figures 4.6c and 4.6f are almost the same as the true ones shown in Figures 4.6a and 4.6d.

Figure 4.7 shows seismic normal vectors estimated for the image with two unconformities shown in Figure 4.4. Both the vertical \((\text{Figure 4.7a})\) and horizontal \((\text{Figure 4.7c})\) components of seismic normal vectors, estimated from structure tensors computed as in equation 4.1, are smooth at the unconformities; those estimated from structure tensors computed as in
equation 4.10 preserve discontinuities at unconformities (Figures 4.7b and 4.7d).

Figure 4.7: From the seismic image as shown in Figure 4.4, the vertical \((u_1)\) and horizontal \((u_2)\) components of seismic normal vectors estimated using structure tensors computed with (b, d) and without (a, c) unconformity constraints.

### 4.4.2 Seismic image flattening at unconformities

Seismic normal vectors or slopes can be used to flatten (Lomask et al., 2006; Parks, 2010) or unfold (Luo and Hale, 2013) a seismic image to generate a horizon volume (Wu and Hale, 2015b), that allows for the extraction of all seismic horizons in the image. However, neither of these methods correctly flatten seismic images with unconformities for two reasons. Firstly, estimated seismic normal vectors or slopes of seismic reflectors are inaccurate at unconformities with multi-oriented structures. Second, seismic reflectors or horizons terminate at unconformities that represent geologic age gaps.

In this paper we have proposed methods to automatically detect unconformities and more accurately estimate seismic normal vectors at unconformities. Therefore, we can easily extend the flattening method described in Wu and Hale (2015b), to better flatten a seismic image at unconformities, by using seismic normal vectors estimated from structure tensors computed with equation 4.10, and by incorporating constraints derived from unconformity likelihoods into the flattening method. We incorporate unconformity constraints in our
flattening method by weighting the equations for flattening using unconformity likelihoods, and then using the unconformity likelihoods to construct a preconditioner in the conjugate gradient method used to solve those equations.

### 4.4.3 Weighting

To generate a horizon volume or to flatten a seismic image, we first solve for vertical shifts \( s(x, y, z) \) as discussed in Wu and Hale (2015b):

\[
\begin{bmatrix}
  w(-\frac{\partial s}{\partial x} - p\frac{\partial s}{\partial z}) \\
  w(-\frac{\partial s}{\partial y} - q\frac{\partial s}{\partial z}) \\
  \epsilon\frac{\partial s}{\partial z}
\end{bmatrix} \approx \begin{bmatrix} wp \\ wq \\ 0 \end{bmatrix}, \tag{4.12}
\]

where \( p(x, y, z) \) and \( q(x, y, z) \) are inline and crossline reflector slopes computed from seismic normal vectors; \( w(x, y, z) \) represent weights for the equations; and the third equation \( \epsilon\frac{\partial s}{\partial z} \approx 0 \), scaled by a small constant \( \epsilon \), is used to reduce rapid vertical variations in the shifts.

For a seismic image with unconformities, we incorporate constraints derived from unconformity likelihoods into the equations 4.12 by setting \( w(x, y, z) = 1 - g_t(x, y, z) \) and we use a spatially variant \( \epsilon(x, y, z) \) instead of a constant value:

\[
\epsilon(x, y, z) = \epsilon_0(1 - g_t(x, y, z)), \tag{4.13}
\]

where \( \epsilon_0 \) is a small constant number (we use \( \epsilon_0 = 0.01 \) for all examples in this paper), and \( g_t(x, y, z) \) denotes the thinned unconformity likelihoods, such as those shown in Figure 4.5b.

This spatially variant \( \epsilon(x, y, z) \), with smaller values (nearly 0) at unconformities, helps to generate more reasonable shifts with gradual variations everywhere except at unconformities.

### 4.4.4 Preconditioner

As discussed in Wu and Hale (2015b), to obtain the shifts \( s(x, y, z) \) in equation 4.12 for a 3D seismic image with \( N \) samples, we solve its corresponding least-squares problem expressed in a matrix form:

\[
(WG)^\top WG s = (WG)^\top W v, \tag{4.14}
\]
where $\mathbf{s}$ is an $N \times 1$ vector containing the unknown shifts $s(x, y, z)$, $\mathbf{G}$ is a $3N \times N$ sparse matrix representing finite-difference approximations of partial derivatives, $\mathbf{W}$ is a $3N \times 3N$ diagonal matrix containing weights $w(x, y, z)$ and $\epsilon(x, y, z)$, and $\mathbf{v}$ is a $3N \times 1$ vector with $2N$ slopes $p$ and $q$, and $N$ zeros.

Because the matrix $(\mathbf{WG})^\top \mathbf{WG}$ is symmetric positive-semidefinite, we can solve the linear system of equation 4.14 using the preconditioned conjugate gradient method, with a preconditioner $\mathbf{M}^{-1}$ as in Wu and Hale (2015b):

$$
\mathbf{M}^{-1} = \mathbf{S}_x \mathbf{S}_y \mathbf{S}_z \mathbf{S}_z^\top \mathbf{S}_y^\top \mathbf{S}_x^\top,
$$

where $\mathbf{S}_x$, $\mathbf{S}_y$ and $\mathbf{S}_z$ are filters that smooth in the $x$, $y$ and $z$ directions, respectively.

For a seismic image with unconformities, the filters $\mathbf{S}_x$, $\mathbf{S}_y$ and $\mathbf{S}_z$ are spatially variant filters designed as in equation 4.11, to preserve discontinuities in shifts $s(x, y, z)$ at unconformities.

Figure 4.8: RGT (a) and flattened (c) images generated with inaccurate seismic normal vectors (Figures 4.7a and 4.7c) and without unconformity constraints. Improved RGT (b) and flattened (d) images with more accurate seismic normal vectors (Figures 4.7b and 4.7d) and constraints from unconformity likelihoods (Figure 4.4).
Figure 4.9: Generated RGT volume (a) and flattened (b) 3D seismic image. Discontinuities in the RGT volume correspond to vertical gaps or hiatuses (blank areas in (b)) in the flattened image at unconformities.
4.4.5 Results

With the computed shifts $s(x, y, z)$, we first generate an RGT volume $\tau(x, y, z) = z + s(x, y, z)$ (Figures 4.8a and 4.9a). We then use the RGT volume to map a seismic image $f(x, y, z)$ (Figures 4.4 or 4.5) in the depth-space domain to a flattened image $\tilde{f}(x, y, \tau)$ (Figures 4.8b or 4.9b) in the RGT-space domain.

From the 2D example shown in Figure 4.8, the RGT (Figure 4.8a) and flattened (Figure 4.8c) images, generated with inaccurate seismic normal vectors (Figure 4.7a and 4.7c) and without unconformity constraints, are incorrect at unconformities, where we expect discontinuities in the RGT image and corresponding gaps in the flattened image. With more accurate seismic normal vectors (Figure 4.7b and 4.7d) and with constraints derived from unconformity likelihoods (Figure 4.4), we obtain an improved RGT image (Figure 4.8b) with discontinuities at unconformities. Using this RGT image, we obtain an improved flattened image (Figure 4.8d), in which seismic reflectors are horizontally flattened and unconformities appear as vertical gaps.

Figure 4.9 shows a 3D example with two unconformity surfaces, highlighted by unconformity likelihoods in Figure 4.5. Using the two unconformity surfaces as constraints, we compute a reasonable RGT volume (Figure 4.9a) with obvious discontinuities at unconformities. We then use this RGT volume to compute a flattened image or 3D seismic Wheeler volume (Figure 4.9b), in which the unconformities are represented as vertical gaps or hiatuses and all seismic reflectors are flattened. The time slice of an RGT image shows large RGT variations between samples in the lower-left and upper-right areas that are separated by an unconformity. This indicates that the sediments, represented by the samples in the two different areas, are deposited in two different sedimentary sequences occurring at different geologic ages.
4.5 Conclusion

We have proposed a method to obtain an unconformity likelihood attribute from the differences between two seismic normal vector fields estimated from two structure-tensor fields, one is computed using a vertically causal smoothing filter, and the other using a vertically anti-causal filter. From a seismic image, we first compute smoothed outer products of image gradients by applying a structure-oriented smoothing filter to each element of these outer products. These smoothed outer products are then smoothed using vertically causal and anti-causal filters to compute two different structure-tensor fields, and their corresponding normal vector fields.

Using structure-oriented smoothing filters for the outer products, we extend structure variations from a termination area to the corresponding parallel unconformity and correlative conformity. In doing this, we assume that the correlative conformity or parallel unconformity is not dislocated by faults. If faults appear in the seismic image, we could perform unfaulting (Luo and Hale, 2013) before attempting to detect unconformities.

We use separate vertically causal and anti-causal filters to obtain structure tensors that differ at unconformities. Unconformity likelihoods might be further improved by instead using causal and anti-causal filters that smooth in directions orthogonal to unconformities.

As examples of applications, we have shown how to estimate more accurate seismic normal vectors and better flatten seismic reflectors at unconformities by using unconformity likelihoods as constraints.

4.6 Acknowledgments

This research is supported by the sponsor companies of the Consortium Project on Seismic Inverse Methods for Complex Structures. All of the real seismic images used in this study are extracted from the F3 seismic data that was graciously provided by dGB Earth Sciences B.V. through OpendTect.
CHAPTER 5
HORIZON VOLUMES WITH CONSTRAINTS

Modified from a paper published in Geophysics, 2015, 80 (2), IM21-IM33
Xinming Wu\textsuperscript{1} and Dave Hale\textsuperscript{1}

5.1 Summary

Horizons are geologically significant surfaces that can be extracted from seismic images. Color-coding of horizons based on amplitude or other attributes can help reveal ancient sedimentary environments and structural features. Extracted horizons are also used for building structure models and stratigraphic interpretations. We propose two methods for constructing seismic horizons aligned with reflectors in a 3D seismic image. The first method generates horizons one at a time; the second generates an entire volume of horizons at once by first computing a relative geologic time volume from seismic normal vectors. Rather than gradually building a horizon by extending one or more seed points to a surface along seismic reflectors, both of our methods generate whole horizons at once by solving partial differential equations derived from seismic normal vectors. The most significant new aspect of both methods is the ability to specify, perhaps interactively during interpretation, a small number of control points that may be scattered throughout a 3D seismic image. Examples show that with our method control points enable the extraction of more accurate horizons from seismic images in which noise, unconformities, and faults are apparent. These points represent constraints that we implement simply as preconditioners in the conjugate gradient method used to construct horizons.

\textsuperscript{1}Center for Wave Phenomena, Department of Geophysics, Colorado School of Mines
5.2 Introduction

In seismic interpretation, by visually tracking or automatically extracting surfaces throughout a 3D seismic image along consistent seismic waveforms, such as peaks, troughs, or zero-crossing points, and to a lesser extent, relatively constant phase, we are able to identify seismic horizons. These horizons are assumed to correspond to stratal surfaces which are primary beddings or ancient depositional surfaces that are geologically synchronous (Vail et al., 1977). Color-coding of horizons based on amplitude or other attributes can help reveal ancient depositional environments and geomorphic features (Posamentier et al., 2007). Therefore, extracting horizons from seismic images is a common and important problem for seismic interpretation.

5.2.1 Horizon volume

Zeng et al. (1998b) first presented the concept of a “stratal time model” and generated such a model with a limited number of interpreted horizons and therefore with limited resolution. A “horizon cube” (de Groot et al., 2010; Qayyum et al., 2012) is a volume containing a dense set of stratigraphic surfaces (Brouwer et al., 2011), which is similar to a “stratal time model” if the surfaces are displayed in geologic time and space domain. Clark et al. (2010a,b) generated a high resolution stratal time model, but called it a “horizon volume”, by using seismic dips estimated from a seismic image. We also prefer to use the term “horizon volume” instead of “stratal time model”, because we also compute a high resolution result (Figure 5.1c) from a high resolution relative geologic time (RGT) volume (Figure 5.1b). The RGT volume is computed from seismic normal vectors which are, similar to seismic dips, estimated from a seismic image (Figure 5.1a). A horizon volume \( t(x, y, \tau) \) (Figure 5.1c) contains the seismic travel time location \( t \) of horizons as a function of RGT \( \tau \) and horizontal spatial coordinates \( x \) and \( y \). Therefore, a horizon volume (Figure 5.1c) can be used to flatten reflectors (Figure 5.1d) or to access all horizons at once. Horizontally slicing a horizon volume yields a travel time structure map of a horizon corresponding to a
constant RGT $\tau$.

The concept of an “RGT volume”, first presented by Stark (2003, 2004, 2005), is closely related to the “horizon volume”. An RGT volume $\tau(x, y, t)$ (Figure 5.1b) contains RGT $\tau$ as a function of spatial coordinates $x, y$ and seismic travel time $t$. A surface of constant $\tau$ in an RGT volume corresponds to a seismic horizon (Stark, 2003, 2005). The only difference between an RGT volume and its corresponding seismic image is that the value of a sample in an RGT volume represents geologic time rather than seismic amplitude (Stark, 2005).
Given an RGT volume $\tau(x, y, t)$ with $\tau$ monotonically increasing with vertical travel time $t$, a horizon volume $t(x, y, \tau)$ can be easily obtained via an inverse linear interpolation method (Parks, 2010) or by a time-warping technique (Burnett and Fomel, 2009). In practice, we use both volumes to conveniently access horizons. An RGT volume, with axes identical to a seismic image, is first used to look up the RGT value $\tau$ for a horizon we wish to extract. A horizon volume is then used to directly obtain the spatial coordinates for the horizon by simply horizontally slicing the horizon volume for that $\tau$. As we compute an RGT value for every seismic sample, we can extract a horizon at each sample in a seismic image, and therefore obtain all seismic horizons represented in a seismic image.

5.2.2 Previous methods

Methods for obtaining a horizon volume can be generally classified into three categories. The first is stratal slicing (Zeng et al., 1998a,b), which uses several reference horizons to interpolate a stratal time model or horizon volume. With a limited number of horizons for control, the interpolated horizon volume can follow large-scale features but usually cannot resolve local features (Lomask et al., 2011).

The second category of methods uses seismic reflector dips (Fomel, 2010; Karimi and Fomel, 2011; Lomask et al., 2006; Parks, 2010) or, similarly, seismic normal vectors, computed for every image sample to be perpendicular to seismic amplitude reflectors. In these methods, a horizon volume is explicitly (Lomask et al., 2006) or implicitly (Fomel, 2010; Karimi and Fomel, 2011; Luo and Hale, 2013; Parks, 2010) generated to map a seismic image from the travel time-space domain to a flattened image in the RGT-space domain. Horizon volumes generated by these methods are more accurate for revealing local features than those interpolated from several horizons using the first category of methods.

The third category is similar to the second one in that these methods also compute high resolution horizon volumes, but without use of dips or normal vectors. Instead, they use an RGT volume generated by unwrapping a corresponding seismic instantaneous phase image (Stark, 2003, 2004, 2005; Wu and Zhong, 2012a).
5.2.3 Proposed methods

We first describe a method for extracting single horizons, one at a time, by using precomputed seismic normal vectors which are perpendicular to seismic amplitude reflectors. This method requires at least one control point to indicate the horizon (containing this point) that we want to extract and to initialize a horizontal surface passing through this point. The initial surface is typically inconsistent with the desired horizon, but it is iteratively deformed until vectors normal to the surface are aligned with vectors normal to a reflector in the seismic image. We extend this method to permit additional control points, which enable reliable extraction of a sequence boundary or a horizon complicated by faults or noise.

We then introduce a second method that generates a complete horizon volume constrained by one or more sets of control points, where each set contains more than one control points. To generate a horizon volume (Figure 5.1c), we first use seismic normal vectors to compute an RGT volume (Figure 5.1b), from which a horizon volume is then interpolated. This process is similar to Parks’s (2010) method for flattening a seismic image, but we instead derive the method in a simpler way. Furthermore, similar to the way in which we extract a more accurate single horizon using control points, we use multiple sets of control points to generate a more accurate horizon volume from a seismic image complicated by faults or noise. Each set of control points belongs to a single horizon with an unspecified RGT value, and is easily specified by simply selecting points that we want to lie on the same horizon. We implement these constraints with simple preconditioners in the conjugate gradient (CG) algorithm we use to compute the RGT and horizon volumes.

5.3 Extracting a single horizon

To extract or construct a single horizon from a 3D seismic image, one usually first picks a reference point or seed. This seed then grows to a horizon surface by manually or automatically tracking seismic reflectors along seismic amplitude peaks or troughs.
Here, we describe a different method that uses at least one control point to initialize a complete horizontal surface and then updates that surface to conform to seismic normal vectors. We then extend this method to enable use of multiple control points, which improve both the accuracy and efficiency of horizon extraction.

5.3.1 Horizon extraction without constraints

We first use structure tensors (Fehmers and Höcker, 2003; Van Vliet and Verbeek, 1995) to compute, for each image sample, a unit (or seismic normal) vector \( \mathbf{n} = [n_x \ n_y \ n_t]^\top \) that is perpendicular to the seismic amplitude reflector at that sample location. We notate the unit vectors by time in the vertical dimension, but we consider a seismic image by samples in both horizontal and vertical dimensions when we estimate those normal vectors. We then assume a single-valued horizon surface \( t = f(x, y) \). The surface can be implicitly defined as a set of points \((x, y, t)\) satisfying

\[
F(x, y, t) = t - f(x, y) = 0. \tag{5.1}
\]

The defined function \( F(x, y, t) \) is 0 everywhere, but its gradient vectors are not zero vectors and can be represented as \( \nabla F(x, y, t) = [-\frac{\partial f}{\partial x} - \frac{\partial f}{\partial y} \ 1]^\top \) where \( \|\nabla F(x, y, t)\| \geq 1 \). The unit vectors perpendicular to the surface are

\[
\mathbf{n}_s = \frac{\nabla F(x, y, t)}{\|\nabla F(x, y, t)\|} = \alpha \begin{bmatrix} -\frac{\partial f}{\partial x} \\ -\frac{\partial f}{\partial y} \\ 1 \end{bmatrix}, \tag{5.2}
\]

where \( \alpha = \frac{1}{\|\nabla F(x, y, t)\|} \) are spatially variant scale factors that make \( \mathbf{n}_s \) unit vectors. Here we assume the surface normal vectors always point downward.

We assume a surface that follows a seismic reflector is a horizon surface of constant geologic time, and seismic normal vectors computed from a seismic amplitude image are unit vectors that are perpendicular to seismic reflectors. Therefore, the seismic horizon we seek is a surface whose normal vectors \( \mathbf{n}_s \) must be equal to the seismic normal vectors \( \mathbf{n} \) at all positions \((x, y, t)\) on the horizon. However, we initially do not know the positions of
the horizon. To solve this problem, we must iteratively update an initial horizontal surface \( f^0(x, y) \), by solving the partial differential equations

\[
\alpha^i \begin{bmatrix} -\frac{\partial f^i}{\partial x} \\ -\frac{\partial f^i}{\partial y} \\ 1 \end{bmatrix} \approx \begin{bmatrix} n_x^{i-1} \\ n_y^{i-1} \\ n_t^{i-1} \end{bmatrix}.
\]

Here, \( f^i(x, y) \) is a surface computed at the \( i \)-th iteration; \( n_x^{i-1} = n_x(x, y, f^{i-1}(x, y)) \), \( n_y^{i-1} = n_y(x, y, f^{i-1}(x, y)) \), and \( n_t^{i-1} = n_t(x, y, f^{i-1}(x, y)) \) are the components of seismic normal vectors at positions on the surface obtained in the \((i-1)-th\) iteration.

To start this iterative process, we initialize a horizontal surface \( f^0(x, y) \) (black lines in Figure 5.2a) passing through a control point (green circle in Figure 5.2a) that is located on the seismic horizon we want to extract. This initial surface is then iteratively updated to align with the seismic horizon. In each iteration, we have \( \alpha^i = n_t^{i-1} \) from the third equation of equations 5.3, and then substitute \( \alpha^i = n_t^{i-1} \) into the first two equations to obtain the following inverse-gradient problem (Bienati and Spagnolini, 2001; Farnebäck et al., 2007) to update the surface \( f^i(x, y) \):

\[
\begin{bmatrix} \frac{\partial f^i}{\partial x} \\ \frac{\partial f^i}{\partial y} \end{bmatrix} \approx \begin{bmatrix} p^{i-1} \\ q^{i-1} \end{bmatrix},
\]

where \( p^{i-1} = -n_x^{i-1}/n_t^{i-1} \) and \( q^{i-1} = -n_y^{i-1}/n_t^{i-1} \) are reflector slopes in the \( x \) and \( y \) directions, respectively. Here, we assume seismic reflectors cannot be vertical and seismic normal vectors always point downward, then \( n_t > 0 \). These two equations above should be satisfied for every sample on the horizon, but it usually helps to weight these equations by some measure \( w(x, y, t) \) of the quality of the estimated reflector slopes. For example, as noise is considered non-planar in general, \( w(x, y, t) \) can be a measure of local planarity in the seismic image, easily computed from structure tensors (Hale, 2009). Then,
\[ w^{i-1} \begin{bmatrix} \frac{\partial f_i}{\partial x} \\ \frac{\partial f_i}{\partial y} \end{bmatrix} \approx w^{i-1} \begin{bmatrix} p^{i-1} \\ q^{i-1} \end{bmatrix}, \quad (5.5) \]

where \( w^{i-1} = w(x, y, f^{i-1}(x, y)) \).

Assuming we have \( N \) sampled locations on the horizon surface, we will have \( 2N \) weighted equations for the \( N \) unknowns \( f^i(x, y) \). For each iteration, we discretize these equations to obtain the corresponding matrix form

\[ WGf \approx Wv, \quad (5.6) \]

where \( W \) is a \( 2N \times 2N \) diagonal matrix containing weights \( w(x, y, f^{i-1}(x, y)) \), \( G \) is a \( 2N \times N \) sparse matrix obtained by discretizing partial derivatives, \( v \) is a \( 2N \times 1 \) vector containing the seismic reflector slopes \( p^{i-1} \) and \( q^{i-1} \) on the surface \( f^{i-1}(x, y) \) obtained in the previous iteration, and \( f \) is an \( N \times 1 \) vector containing surface depths \( f^i(x, y) \) we want to find.

We use approximate equalities in equations 5.3, 5.4, 5.5 and 5.6 because we compute the least-squares solution by solving the normal equation of equation 5.6

\[ (WG)^\top WGf = (WG)^\top Wv. \quad (5.7) \]

To simplify this equation, we let \( A = (WG)^\top WG \) and \( b = (WG)^\top Wv \) to obtain

\[ Af = b. \quad (5.8) \]

Since the matrix \( A = G^\top W^\top WG \) is symmetric positive definite (SPD), we use the CG method to solve this linear system.

### 5.3.2 Preconditioner

To accelerate the convergence of CG iterations, we use the model reparameterization technique \( f = S\tilde{f} \) (Fomel and Claerbout, 2003; Harlan, 1995; VanDecar and Snieder, 1994). \( S \) is a simplification operator designed to create the desired features in the solution \( \tilde{f} \). Applying this technique to the system of equation 5.8, we first solve a new system

\[ S^\top AS\tilde{f} = S^\top b \quad (5.9) \]
for the new unknowns $\tilde{f}$ and then compute the desired solution $f = S\tilde{f}$. For an appropriate operator $S$, the CG method applied to the new system of equation 5.9 converges much faster than for the original system of equation 5.8.

In effect, this model reparameterization is equivalent to *split preconditioning* (Saad, 1996) with left and right preconditioners $M_L^{-1} = S^T$ and $M_R^{-1} = S$. As noted by Saad (1996), this split preconditioning can be implemented with a left preconditioning matrix $M = M_L M_R$ in a preconditioned CG solution of

$$M^{-1}Af = M^{-1}b,$$

where $M^{-1} = SS^T$.

Recall that $S$ is a simplification operator used to facilitate desired features in the solution (Harlan, 1995). Here, we implement $S$ as a smoothing operator $S = S_x S_y$, where $S_x$ and $S_y$ are axis-aligned smoothing filters in the $x$ and $y$ directions, respectively. A horizon surface $f$ is often smooth, except at faults. Therefore, our $S_x$ and $S_y$ are spatially variant smoothing filters (Hale, 2009), with the extent of smoothing controlled by a measure of discontinuity of seismic reflectors. This measure could be planarity (Hale, 2009) or fault likelihood (Hale, 2013b). Here we use planarity, computed from structure tensors, to control the extent of smoothing in $S_x$ and $S_y$.

Now, for each iteration (equation 5.5) that updates the surface $f^i(x, y)$, we solve equation 5.8 using the preconditioned CG method with preconditioner

$$M^{-1} = S_x S_y S_y^T S_x^T.$$  

In this way, we iteratively update the surface $t = f(x, y)$ until its normal vectors $n_s$ are aligned with the seismic normal vectors $n(x, y, t = f(x, y))$. The updating iteration is terminated when the absolute average update of each sample on the surface is smaller than some small number

$$\frac{\sum_{j=0}^{N-1} |f^i(x_j, y_j) - f^{i-1}(x_j, y_j)|}{N} < \epsilon_t,$$

where $N$ is the number of samples on the surface, and $\epsilon_t$ is an arbitrary small number.
In summary, given an initially horizontal surface (black curves in Figure 5.2) that is inconsistent with any seismic reflector, our method iteratively reduces the difference between the normal vectors $\mathbf{n}_s$ of the surface and the seismic normal vectors $\mathbf{n}(x, y, f(x, y))$ on the surface to obtain a single seismic horizon surface (blue curves in Figure 5.2).

Figure 5.2: Seismic sections (a) and subsections (b) that intersect with a sequence boundary. The initially horizontal surface (black curve) passes through one control point and is updated iteratively using seismic normal vectors. The dashed green curve denotes the manually interpreted sequence boundary.

### 5.3.3 Results without constraints

In Figure 5.2, using only one control point to indicate which horizon we want to extract, our method updates the initially horizontal surface to the more nearly correct seismic horizon (blue curves in Figure 5.2) after 9 iterations. The extracted surface is well-aligned with the seismic horizon at conformable areas in the left section of Figure 5.2a. However, in the sections shown in Figure 5.2b, this iterative method fails to update the horizon surface to the location of the angular unconformity (green dashed curve in Figure 5.2b).

Extracting such a sequence boundary or unconformity is an important but difficult problem in seismic interpretation. From structure tensors, we fail to correctly estimate the discontinuous normal vectors at the unconformity and therefore obtain the incorrect horizon surface shown in Figure 5.2b. In the next section, we describe a method to more accurately...
extract a sequence boundary using control points.

5.3.4 Horizon extraction with constraints

Near unconformities, faults, or in areas where an image is noisy, estimated seismic normal vector are not accurate enough to automatically obtain a correct sequence boundary or horizon. Therefore, instead of using a fully automatic method, we might manually interpret the seismic image to obtain a more geologically reasonable surface. However, we need not manually interpret the entire horizon. Using a small number of control points as constraints, we solve a constrained least-squares problem to efficiently and more accurately extract a sequence boundary or horizon from a noisy or complex seismic image.

5.3.5 Constrained optimization

As discussed above, in each iteration that updates a horizon surface, we solve a linear system \( \mathbf{A} \mathbf{f} = \mathbf{b} \) for the vector \( \mathbf{f} \) that represents the surface. Because the matrix \( \mathbf{A} \) is SPD, solving this linear system is equivalent to minimizing the following quadratic function of the vector \( \mathbf{f} \):

\[
F(\mathbf{f}) = \frac{1}{2} \mathbf{f}^\top \mathbf{A} \mathbf{f} - \mathbf{b}^\top \mathbf{f}.
\] (5.12)

Suppose we have a set of \( n \) control points \( (x_i, y_i, t_i), \) \( i = 1, 2, ..., n \), and we want to extract a horizon surface that exactly passes through these points. With these constraints, we obtain a constrained optimization problem:

\[
\text{minimize} \quad F(\mathbf{f}) = \frac{1}{2} \mathbf{f}^\top \mathbf{A} \mathbf{f} - \mathbf{b}^\top \mathbf{f},
\]

subject to

\[
\mathbf{C} \mathbf{f} = \mathbf{t},
\]

(5.13)

where \( \mathbf{t} = [t_1, t_2, ..., t_n]^\top \) is an \( n \times 1 \) column vector, and \( \mathbf{C} \) is an \( n \times N \) (where, again, \( N \) is the number of samples on the surface) sparse matrix with ones at the positions corresponding to control points and zeros elsewhere. Assuming we have found some solution \( \mathbf{f}_0 \) to the constraint equation \( \mathbf{C} \mathbf{f}_0 = \mathbf{t} \), and a matrix \( \mathbf{Z} \) whose columns form a basis for the null space
of $C$ so that $CZ = 0$, then any solution $f$ of the constraint equation $Cf = t$ can be written as

$$f = f_0 + Zp,$$  \hspace{1cm} (5.14)

where $p$ is a reduced $(N - n) \times 1$ column vector, and again $n$ is the number of control points. The control points must be unique to ensure that the matrix $C$ has $n$ linearly independent rows and $Z$ has $N - n$ linearly independent columns.

Substituting equation 5.14 into equation 5.12, we obtain a quadratic function $F(p)$ with the reduced vector $p$:

$$F(p) = \frac{1}{2}(f_0 + Zp)^\top A(f_0 + Zp) - b^\top(f_0 + Zp).$$  \hspace{1cm} (5.15)

Minimizing this quadratic function for the reduced solution $p$ is equivalent to solving the following reduced linear system

$$Z^\top AZp = Z^\top(b - Af_0).$$  \hspace{1cm} (5.16)

We can now solve this reduced system to get $p$, and then recover the desired solution $f$ by using equation 5.14.

5.3.6 Constrained preconditioner

Before we can solve equation 5.16, we must find matrix $Z$ and vector $f_0$. Fortunately, these subproblems are simple. For example, assume we have three control points: $f_0 = t_0$, $f_2 = t_2$, and $f_3 = t_3$, then $t = [t_0 \ t_2 \ t_3]^\top$ and the matrix $C$ is

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{3 \times N}. \hspace{1cm} (5.17)$$

We can immediately find a solution $f_0 = [t_0 \ 0 \ t_2 \ 0 \ \cdots \ 0]^\top$ to the constraint equation $Cf_0 = t$. The columns of matrix $Z$ form a basis of the null space of matrix $C$, so that $CZ = 0$. We generate such a matrix $Z$ from an $N \times N$ identity matrix, by simply removing any columns that are identical to rows in the matrix $C$: 89
Given $Z$ and the solution $f_0$, we are ready to solve the reduced system shown in equation 5.16. Because the matrix $Z^TAZ$ is SPD, we can use the CG method to solve this reduced system. Many authors (e.g., Dollar, 2005; Gould et al., 2001; Nash and Sofer, 1996) have discussed the solution of this system using the preconditioned CG method, and we use a simple preconditioner $P_z$ described in Nash and Sofer (1996):

$$P_z = Z^TM^{-1}Z \approx (Z^TAZ)^{-1},$$

where $M^{-1} = S_xS_yS_y^TS_x^T$ as in equation 5.11, and $Z^TZ = I$ since the columns of $Z$ form a basis. Therefore, our preconditioner for the reduced system is

$$P_z = Z^TS_xS_yS_y^TS_x^TZ.$$  

In the preconditioned CG method for the reduced system, one would compute the initial residual $r_z = Z^T(b - Af_0) - Z^TAZp$ and the preconditioned residual $g_z = P_zr_z$. Instead of solving the reduced system to obtain $p$ and then recovering the desired solution $f$, we can instead directly solve for $f$ because we have a relationship between the reduced and full solutions $f = f_0 + Zp$. As discussed by Gould et al. (2001), to explicitly perform the multiplication by $Z$ and the addition of the term $f_0$ in the CG method, we may choose $f = f_0 + Zp$, $Z^Tr = r_z$ and $g = Zg_z$, so that $g = ZP_zZ^Tr$. This process is equivalent to applying the preconditioned CG method to the unconstrained linear system $Af = b$, with a preconditioner

$$P = ZP_zZ^T = ZZ^TM^{-1}ZZ^T = ZZ^TS_xS_yS_y^TS_x^TZ.$$  

(5.21)
In practice, we do not explicitly form the matrices $A$ and $ZZ^\top$ because the preconditioned CG method requires only the computation of the residual vector $r = b - Af$ and gradient vector $g = Pr$.

It is trivial to compute vector $ZZ^\top x$ for any $N \times 1$ vector $x$ because $ZZ^\top$ has the form

$$ZZ^\top = \begin{bmatrix}
0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1
\end{bmatrix}_{N \times N} \tag{5.22}$$

Computation of $ZZ^\top x$ simply zeros all the elements of $x$ with indices corresponding to the locations of control points.

With the preconditioner $P$ denoted by equation 5.21, the preconditioned gradient $g = Zg_z = ZP_zZ^\top r$ is projected to be in the null space of $C$. As a result, all updates to the solution $f$ in this preconditioned CG method will also lie in the null space of $C$. Therefore, as the initial solution $f_0$ satisfies the constraints $Cf_0 = t$, the solution $f$ after each CG iteration also satisfies $Cf = t$.

5.3.7 Results with constraints

Where seismic normal vectors estimated from structure tensors are inaccurate (e.g., near unconformities, faults and noisy data), the use of control points helps to extract a more reliable horizon or sequence boundary. As shown in Figure 5.2, when we extract a sequence boundary constrained by only one control point (green circle in Figure 5.2a), the surface we extract (blue curves in Figure 5.2) is well aligned with a seismic reflector in the conformable areas (the left-side section and the left part of the right-side section in Figure 5.2a), where seismic normal vectors can be estimated accurately. However, the surface (blue curves) extracted at the unconformity (Figure 5.2b) deviates from the manually interpreted surface (dashed green curve in Figure 5.2b) because the normal vectors estimated there are inaccurate.
Using 19 control points (green points in Figure 5.3c), we obtain a surface that better fits the manually interpreted sequence boundary. Figure 5.3a shows crossline and inline seismic sections that intersect the sequence boundaries extracted using (1) only one control point (blue curves), and (2) 19 control points (green curves). We observe that the sequence boundary extracted using 19 control points better represents the manually interpreted unconformity surface compared to the one extracted using only one control point. Figures 5.3b and 5.3c show the same extracted sequence-boundary surfaces colored with seismic ampli-
tudes. Amplitude values for 19 control points (Figure 5.3c) are more uniform than those for one control point (Figure 5.3b).

This sequence boundary is also complicated by faults, highlighted by red ellipses in Figure 5.3. The surface extracted using only 1 control point (blue curves in Figure 5.3a and the surface in Figure 5.3b) is inaccurate near faults. However, the surface with 19 control points (green curves in Figure 5.3a and the surface in Figure 5.3c) more accurately follows the faults. This example demonstrates that constraints facilitate extraction of a horizon surface complicated by faults.

Moreover, with more control points, an initial surface converges more quickly to the final extracted horizon. We can use more control points to interpolate a better initial surface \( f^0(x, y) \) that is smooth but exactly passes through the control points. An initial surface interpolated using more control points will be closer to the seismic horizon \( f(x, y) \) we want to extract, which therefore enables the CG method to more quickly converge to that horizon. For example, it takes nine iterations to converge using one control point (blue curves in Figure 5.3a), but only five iterations to converge using 19 control points (green curves in Figure 5.3a).

### 5.4 Generating a horizon volume

Using the method discussed above, we can extract a single seismic horizon or sequence boundary with one or more control points that represent interpreted constraints. With similar constraints, we can also extract all seismic horizons from a seismic image at once, and thereby generate a complete horizon volume. In a horizon volume \( t(x, y, \tau) \), as shown in Figure 5.1c, the vertical axis is RGT \( \tau \) and color denotes seismic travel time \( t \). Horizontally slicing a horizon volume at any single RGT value \( \tau \) yields a seismic horizon.

Here, we first describe a method for using seismic normal vectors to automatically generate a horizon volume without constraints, which is usually accurate for seismic images with simple structures. To better handle images complicated by faults or noise, we then extend this method, by incorporating scattered sets of interpreted points that correspond
to multiple seismic horizons, to generate a more reliable horizon volume that honors those interpreted constraints.

![Figure 5.4: The same RGT volume (a) as shown in Figure 5.1b, contours (b) of the RGT volume are horizons in the corresponding seismic image.](image)

5.4.1 Horizon volume without constraints

As discussed by Parks (2010), a horizon volume $t(x, y, \tau)$ can be generated from an RGT volume $\tau(x, y, t)$ by inverse linear interpolation if we assume that $\tau$ in the RGT volume increases monotonically with seismic travel time $t$. Some authors have described methods to generate such an RGT volume using phase unwrapping (e.g., Stark, 2003, 2004; Wu and Zhong, 2012b) or reflector dips (Fomel, 2010; Parks, 2010). Here we rederive the method of Parks (2010) in a simpler way to compute an RGT volume.

In an RGT volume $\tau(x, y, t)$ like that shown in Figure 5.4a or 5.1b, contours (Figure 5.4b) of constant $\tau$ represent seismic horizons, which means these contours should have the same structures as seismic reflectors in the seismic image (Figure 5.4b). Therefore, gradient vectors for an RGT volume $\tau(x, y, t)$, that are perpendicular to RGT contours, should be parallel to seismic normal vectors $\mathbf{n} = [n_x, n_y, n_z]^\top$, that are perpendicular to seismic amplitude reflectors. If we assume that these vectors always point downward, we have
where $\alpha$ is a positive and spatially variant scalar number. Because we again have more equations than unknowns, in general we can only approximately solve these coupled partial differential equations. As we assume that all the seismic normal vectors are always point downward, which means that the vertical component $n_t$ of the normal vectors are always positive ($n_t > 0$). Therefore, RGT results computed using the partial differential equations above usually increase vertically with travel time.

Using the third equation of equations 5.23, we compute $\alpha = (\partial \tau/\partial t)/n_t$, where $n_t > 0$. Substituting $\alpha$ into the first two of equations 5.23, we obtain

\[
\begin{bmatrix}
  n_t \frac{\partial \tau}{\partial x} - n_x \frac{\partial \tau}{\partial t} \\
  n_t \frac{\partial \tau}{\partial y} - n_y \frac{\partial \tau}{\partial t}
\end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\] (5.24)

In attempting to solve these equations, we would need to carefully choose boundary conditions to avoid obtaining the trivial solution $\tau = \text{constant}$.

To avoid this problem, as discussed by Parks (2010), we rewrite $\tau(x, y, t)$ as

\[
\tau(x, y, t) = z + s(x, y, t),
\] (5.25)

where the function $s(x, y, t)$ represents vertical shifts. Substituting equation 5.25 into equation 5.24, we obtain

\[
\begin{bmatrix}
  n_t \frac{\partial s}{\partial x} - n_x \frac{\partial s}{\partial t} \\
  n_t \frac{\partial s}{\partial y} - n_y \frac{\partial s}{\partial t}
\end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \end{bmatrix},
\] (5.26)

or

\[
\begin{bmatrix}
  -\frac{\partial s}{\partial x} - p \frac{\partial s}{\partial t} \\
  -\frac{\partial s}{\partial y} - q \frac{\partial s}{\partial t}
\end{bmatrix} \approx \begin{bmatrix} p \\ q \end{bmatrix},
\] (5.27)
where again $p = -n_x/n_t$ and $q = -n_y/n_t$ are estimated inline and crossline slopes of seismic reflectors. Equation 5.27 is what Parks (2010) solved to obtain shifts that flatten a seismic image.

As suggested by Lomask et al. (2006), we add a third equation $\epsilon s_t \approx 0$ to reduce vertical variations in the shifts. We also weight the equations above by a measure $w(x, y, t)$ of the quality of the estimated reflector slopes $p(x, y, t)$ and $q(x, y, t)$. We then compute the shifts by solving the following equations:

$$
\begin{bmatrix}
  w(-\frac{\partial s}{\partial x} - p\frac{\partial s}{\partial t}) \\
  w(-\frac{\partial s}{\partial y} - q\frac{\partial s}{\partial t}) \\
  \epsilon \frac{\partial s}{\partial t}
\end{bmatrix} \approx \begin{bmatrix} wp \\
  wq \\
  0
\end{bmatrix}.
$$

(5.28)

If we have $N$ image samples, then equation 5.28 represents $3N$ equations for $N$ unknown shifts, and these equations can be expressed in matrix form as

$$
WG_s \approx Wv,
$$

(5.29)

where $s$ is an $N \times 1$ vector containing the unknown shifts $s(x, y, t)$, $G$ is a $3N \times N$ sparse matrix representing finite-difference approximations of partial derivatives, $W$ is a $3N \times 3N$ diagonal matrix containing weights $w(x, y, t)$ and the constant $\epsilon$, and $v$ is a $3N \times 1$ vector with $2N$ slopes $p$ and $q$, and $N$ zeros.

From equation 5.23 to equation 5.29, we use the approximate equalities because we compute the least-squares solution by solving the normal equation of equation 5.29

$$
(WG)^T WG_s = (WG)^T Wv.
$$

(5.30)

Let $A = (WG)^T WG$ and $b = (WG)^T Wv$ so that this linear system becomes

$$
As = b.
$$

(5.31)

The matrix $A$ is both SPD and sparse. In practice, we do not explicitly form the matrices $A$, $W$, and $G$. Instead, we solve this linear system using the CG method, which requires only the computation of matrix-vector products like $As = (WG)^T WG_s$ and $b = (WG)^T Wv$. 

96
As when extracting a single seismic horizon, we solve equation 5.31 using the preconditioned CG method with a preconditioner defined by

\[ M^{-1} = S_x S_y S_t S_t^\top S_y^\top S_x^\top, \]

where, again, \( S_x, S_y \) and \( S_t \) are filters that smooth in the \( x, y \) and \( t \) directions, respectively. We again expect the solution to be laterally smooth except at faults, as shown in Figure 5.6b. Therefore, the lateral smoothing filters \( S_x \) and \( S_y \) are spatially variant filters (Hale, 2009), and the extent of smoothing is proportional to a measure of reflector continuity, so that these filters smooth less at faults. We expect the shifts to be vertically smooth because we assume that there are no unconformities in this example. Therefore, our vertical smoothing filter \( S_t \) in this example is spatially invariant.

![Figure 5.5: A seismic image (a), generated horizon volume (b), and flattened image (c) without control points.](image)

We derive all of the equations above for 3D images, but they can be easily modified to work for 2D images, by simply omitting the second equation for the \( y \) direction from...
equation 5.28. For the 2D example shown in Figure 5.1, we first solved equation 5.31 to get shifts \( s(x,t) \). We then computed an RGT volume \( \tau(x,t) = t + s(x,t) \) (Figure 5.1b), where \( \tau \) increases monotonically (\( \tau \) does not decrease, but increases with different rates) with seismic travel time \( t \). Finally we computed a horizon volume \( t(x,\tau) \) (Figure 5.1c) from the RGT volume \( \tau(x,t) \) by inverse linear interpolation (Parks, 2010). This horizon volume \( t(x,\tau) \) maps the seismic image (Figure 5.1a) to a flattened image (Figure 5.1d).

For seismic images with simple geologic structures and little noise, as in Figure 5.1a, we can use the method discussed above to compute an accurate RGT volume (Figure 5.1b). A horizon volume (Figure 5.1c) is then interpolated from the RGT volume and subsequently used to flatten the input seismic image (Figure 5.1a) to produce the flattened image (Figure 5.1d). However, for seismic images complicated by faults, as in Figure 5.5a, the generated horizon volume (Figure 5.5b) is inaccurate, so that seismic reflectors are not flattened correctly (Figure 5.5c). Therefore, we extend this method to compute more accurate RGT and horizon volumes by incorporating one set or multiple sets of interpreted control points that may correspond to one or multiple horizons, without defining any RGT values for any control points.

### 5.4.2 Horizon volume with constraints

For specified sets of control points, we solve a constrained optimization problem similar to that we solve when extracting a single seismic horizon:

\[
\begin{align*}
m\text{inimize}_s & \quad F(s) = \frac{1}{2} s^\top A s - b^\top s, \\
\text{subject to} & \quad C s = d.
\end{align*}
\] (5.33)

As when extracting a single horizon, solving the constrained problem above is equivalent to solving a corresponding unconstrained problem \( A s = b \) using a preconditioned CG method with an initial solution \( s_0 \) to the constraint equation \( C s_0 = d \) and a constrained preconditioner \( P = Z Z^\top M^{-1} Z Z^\top \), where \( M^{-1} = S_x S_y S_t S_{xt}^\top S_{yt}^\top S_{xt}^\top \). Therefore, to solve this problem, we need only an initial solution \( s_0 \) and the matrix \( Z Z^\top \) for the preconditioner \( P \).
Let us use a tiny 3D seismic image with only $N = 2 \times 2 \times 2$ samples to explain how to implement multiplication by the matrix $ZZ^\top$ and to find an initial solution $s_0$. As in equation 5.25, we want to compute a 3D RGT volume $\tau(x,y,t)$ with shifts $s(x,y,t)$. In this simple example, both $\tau$ and $s$ have only $N = 2 \times 2 \times 2$ samples, and we can express equation 5.25 in vector form as

$$\tau = t + s,$$  \hspace{1cm} (5.34)

where

$$t = [t_0 \ t_1 \ t_2 \ t_3 \ t_4 \ t_5 \ t_6 \ t_7]^\top,$$

$$s = [s_0 \ s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6 \ s_7]^\top,$$

$$\tau = [\tau_0 \ \tau_1 \ \tau_2 \ \tau_3 \ \tau_4 \ \tau_5 \ \tau_6 \ \tau_7]^\top,$$  \hspace{1cm} (5.35)

Assume that we have 2 sets of constraints: the first set has 3 control points with sample indices $\{3, 5, 7\}$, and the second set has 2 control points with sample indices $\{1, 6\}$. Within each set of constraints, all control points are interpreted to be on a single seismic horizon. Therefore, we have $\tau_3 = \tau_5 = \tau_7$ and $\tau_1 = \tau_6$. According to equation 5.34, this means that $s_5 - s_3 = t_3 - t_5$ and $s_7 - s_3 = t_3 - t_7$, and $s_6 - s_1 = t_1 - t_6$. We can therefore write the constraint equation $Cs = d$ as follows:

$$\begin{bmatrix}
0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\
0 & -1 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
s_0 \\
s_1 \\
s_2 \\
s_3 \\
s_4 \\
s_5 \\
s_6 \\
s_7
\end{bmatrix}
= 
\begin{bmatrix}
t_3 - t_5 \\
t_3 - t_7 \\
t_1 - t_6
\end{bmatrix},$$  \hspace{1cm} (5.36)

where, again, $s = [s_0 \ s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6 \ s_7]^\top$. Here, we want to emphasize that we do not specify any RGT values or shifts for the interpreted control points to constrain the generation of an RGT or horizon volume. We only set the RGT values of the control points belong to a same horizon to be equal to construct the constraint equation in 5.36. This makes it easy for an interpreter to incorporate control points for generating a more reliable horizon volume.

In this example, matrix $C$ has 3 linearly independent rows so that matrix $Z$ must have $N - 3$ linearly independent columns, such that $CZ = 0$, because the columns of matrix $Z$ form a basis for the null space of $C$. Construction of matrix $Z$ is only slightly more
complicated than for the single-horizon case. Specifically,

\[ Z = [e_{c_1} \mid e_{c_2} \mid e_0 \mid e_2 \mid e_4] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}_{8 \times 5}, \tag{5.37} \]

where \( e_{c_1} = e_3 + e_5 + e_7 \), \( e_{c_2} = e_1 + e_6 \), and \( e_i \), for \( i = 0, 1, \ldots, N - 1 \), is an \( N \times 1 \) unit vector with 1 at the \( i - \text{th} \) index. In other words, we begin with the identity matrix and simply sum the unit vectors \( e_i \) with indices \( i \) in \( \{3, 5, 7\} \), corresponding to the first set of control points, to obtain the first column of \( Z \); similarly, we obtain the second column of \( Z \), corresponding to the second set of control points with indices \( \{1, 6\} \); and finally, we use all of the remaining unit vectors \( e_i \) that do not correspond to any control point for remaining columns of \( Z \). In the same way, we can easily construct matrix \( Z \) for any number of sets of control points.

We can normalize the columns of matrix \( Z \) to obtain

\[ Z = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 \end{bmatrix}_{8 \times 5}, \tag{5.38} \]

with columns that form an orthonormal basis for the null space of matrix \( C \). Also we have

\[ ZZ^\top = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \end{bmatrix}_{8 \times 8}. \tag{5.39} \]
For any vector $\mathbf{x} = [x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7]^\top$, it is easy to compute the product

$$
ZZ^\top \mathbf{x} = [x_0 \ x_{c_2} \ x_2 \ x_{c_1} \ x_4 \ x_{c_2} \ x_5 \ x_{c_1}]^\top,
$$

(5.40)

where $x_{c_1} = (x_3 + x_5 + x_7)/3$ and $x_{c_2} = (x_1 + x_6)/2$. In other words, we compute $ZZ^\top \mathbf{x}$ by simply gathering and averaging all elements of $\mathbf{x}$ with indices corresponding to each set of control points, and then scattering the averages back into those same elements. In each CG iteration, when we apply the constrained preconditioner $\mathbf{P} = ZZ^\top S_x S_y S_t S_t^\top S_y^\top S_x^\top ZZ^\top$ to a vector, we need only compute averages and apply smoothing filters.

We can also easily find an initial solution $s_0$ to the constraint equation $C s_0 = \mathbf{d}$:

$$
s_0 = [0 \ s_1 \ 0 \ s_3 \ 0 \ s_5 \ s_6 \ s_7]^\top,
$$

(5.41)

in which elements with indices corresponding to the first set of control points are $s_3 = 0$, $s_5 = t_3 - t_5$ and $s_7 = t_3 - t_7$; elements corresponding to the second set of control points are $s_1 = 0$ and $s_6 = t_1 - t_6$. Therefore, to construct an initial set of shifts $s_0$, we use zeros for elements that do not correspond to any control points; for each set of control points, we first compute their average depth or travel time, and then choose the point with depth nearest to that average as the reference point and set it with zero shift (e.g., $s_3 = 0$ for the first set of control points, and $s_1 = 0$ for the second set of control points), then use the depth differences between the reference point and other control points for the remaining initial shifts in $s_0$.

With an initial solution $s_0$ and the constrained preconditioner $\mathbf{P} = ZZ^\top M^{-1} ZZ^\top$, we can apply the preconditioned CG method to the unconstrained system $A s = \mathbf{b}$ to obtain a solution $s$ that satisfies the constrained problem of equation 5.33. In each CG iteration, we compute a residual as $r = \mathbf{b} - A s$. Using the constrained preconditioner $\mathbf{P}$, we compute a constrained residual $r_P = ZZ^\top M^{-1} ZZ^\top r$ that is in the null space of the constraint matrix $C$. This means that all of the updates to the initial solution $s_0$ in this preconditioned CG method will also be in the null space of $C$. Therefore, because the initial solution $s_0$ satisfies the constraint equation $C s_0 = \mathbf{d}$, the final solution $s$ obtained after any number of CG iterations will also satisfy the constraints.
Figure 5.6: A seismic image (a) with 3 pairs of interactively interpreted control points (yellow circles, pluses and squares), generated horizon volume (b) and flattened image (c).

Figure 5.6 is a 2D example that shows how constraints help to generate a more accurate horizon volume and better flatten a seismic image. In this example, we use the same input seismic image (Figure 5.6a) complicated by faults that is displayed in Figure 5.5a, but now we have 3 sets of constraints. For each set of constraints, we interpret 2 control points (yellow circles, pluses, and squares in Figure 5.6a) for each seismic horizon. Using 3 sets of constraints, we compute a more accurate horizon volume (Figure 5.6b), with which we can better flatten (Figure 5.6c) seismic reflectors across faults.

5.4.3 3D results with constraints

Figure 5.7a shows a 3D seismic image that is also complicated by faults. To flatten this 3D image or generate a horizon volume, we choose weights $w(x, y, t)$ corresponding to faults in equation 5.28. Specifically, we use the method developed by Hale (2013b) to first compute an image of fault likelihoods $f(x, y, t) \in [0, 1]$ in which values near 1 indicate fault locations.
Figure 5.7: Input seismic image (a) and a corresponding RGT volume (b) computed with three sets of control points.
Figure 5.8: The flattened seismic image is sliced at $\tau = 1.664$ (a) and $\tau = 1.751$ (b). Horizontal slices in a flattened image correspond to seismic horizon surfaces (upper-right panels in (a) and (b), for which color denotes depth) in an unflattened image.
Figure 5.9: The RGT volume shown in Figure 5.7b is converted to a complete horizon volume (a), in which horizontally slicing (h1∼h6) at six different RGT values yields six seismic horizons displayed in (b). The cut-away views of these six horizons are shown in (c) to reveal more details.
We then use $w = (1 - f)^8$ as weights in equation 5.28, where the power 8 is an arbitrary number to increase the contrast between low and high fault likelihoods.

For this example, we use three sets of constraints, corresponding to three horizons in the 3D seismic image, to compute a more accurate horizon volume and more accurately flatten the seismic image. The first set contains 5 control points, the second one contains 7 control points (green points in Figure 5.8a), and the third one contains 11 control points (green points in Figure 5.8b). Using these three sets of constraints, we first compute an RGT volume as shown in Figure 5.7b, from which we then interpolate a horizon volume (Figure 5.9a) that flattens (Figure 5.8a or 5.8b) seismic reflectors across faults. The constraints help to flatten not only reflectors passing through the control points, but also other reflectors in the 3D seismic image as well.

Figure 5.9a displays the horizon volume computed from the RGT volume shown in Figure 5.7b. Each horizontal slice in the horizon volume is a travel-time structure map of a horizon corresponding to a constant RGT value. Figure 5.9b shows a 3D view of 6 seismic horizons extracted by horizontally slicing (h1∼h6 in Figure 5.9a) the horizon volume at 6 different RGT values. In Figure 5.9b, different colors denote different seismic horizons corresponding to the horizontal slices (h1∼h6) with different colors in Figure 5.9a, but deeper horizons are obscured by the top one. We therefore, in Figure 5.9c, display cut-away views of each of the horizons. We observe that the horizons with control points (the cyan and yellow surfaces) and others without control points coincide well with seismic reflectors.

5.5 Conclusion

We propose methods to (1) extract one seismic horizon at a time and (2) to compute at once a complete horizon volume. We designed these two methods to compute horizons that honor interpreted constraints, specified as sets of control points. We incorporate the control points with simple constraint preconditioners in the CG method used to compute horizons.

The first method is useful, even though we can extract all horizons at once using the second method, because it can more quickly extract a single horizon. Using multiple con-
control points, this method can reliably extract complicated geologic surfaces such as sequence boundaries and horizons with faults. Furthermore, this first method might be used to efficiently extract horizons that might serve as control surfaces (large sets of control points) for the second method.

The second method generates a complete horizon volume at once. With a small number of interpreted constraints, this method works well for seismic images complicated by faults.

Interpreted constraints are necessary, because completely automatic interpretation cannot yet handle complicated seismic horizons. The proposed methods provide an especially simple way to specify such constraints by simply interactively picking points in a 3D seismic image that belong to the same seismic horizon. These methods can be implemented to interactively add or move control points, while quickly updating a single seismic horizon or complete horizon volume.

One minor defect of both methods is that they do not automatically produce gaps representing heaves at faults in an extracted horizon, but a post-processing can be added to detect possible fault positions from the discontinuities of the extracted horizon surface, and then create such gaps at the detected faults. These methods might be further improved if we could predict areas where control points are required to generate more reliable results so that the interpretation of constraints could be more straightforward and efficient.

5.6 Acknowledgments

This research is supported by the sponsor companies of the Consortium Project on Seismic Inverse Methods for Complex Structures. The real seismic images in the 2D examples shown in Figures 1, 4, 5, and 6 are provided by the U.S. Department of Energy. The real seismic images in the 3D examples shown in Figures 2, 3, 7, 8, and 9 are provided by dGB Earth Sciences.
CHAPTER 6
CONCLUSIONS AND SUGGESTIONS

In this thesis, I have proposed enhanced methods to automatically interpret fault, unconformity, and horizon surfaces from 3D seismic images. The main results of this thesis and suggestions for future research are summarized as follows.

6.1 Fault interpretation

In Chapter 2, I propose to use a simple linked data structure to represent a fault surface, to extract multiple intersecting fault surfaces without holes at intersections, and to accurately estimate fault slips. Using this simple data structure, fault surfaces are represented by sets of linked fault samples. Each fault sample corresponds to one and only one seismic image sample, and is displayed as a square colored by fault likelihood, and oriented by fault strike and dip. These linked fault samples can be easily displayed as a 3D fault image (with non-null values only at positions where faults exist) because the samples are located on the grid points of the seismic image. These linked fault samples can also be displayed as a fault surface by simply increasing sizes of squares used to represent fault samples, so that the squares overlap and appear to form a opaque and continuous surfaces. These fault surfaces, however, are not triangle or quad meshes, which are unnecessarily complicated for subsequent processing tasks, such as fault slip estimation.

To construct such a fault surface with this data structure, I first compute fault samples from fault images of fault likelihoods, fault strikes, and fault dips. I then link each fault sample and its above, below, left, and right neighbors with similar fault likelihoods, strikes, and dips. For fault samples with missing neighbors, I propose a method to try to create these neighbors, so as to construct more complete fault surfaces without holes, even when faults intersect. This is important for fault slip estimation, because holes in fault surfaces make it difficult to determine which samples or reflectors of the seismic image should be
used for the dynamic warping method to estimate fault slip vectors. Moreover, holes make
it difficult for the dynamic warping method to enforce the constraints that fault throws (the
vertical components of slips) vary smoothly.

In Chapter 3, I propose two methods to use precomputed fault positions and slips to
compute unfaulting shifts that undo the faulting in a seismic image. In these unfaulting
methods, I construct unfaulting equations for samples adjacent to faults using the estimated
fault positions and slips, so that unfaulting shifts computed for these samples confirm to the
estimated fault slips. For samples away from faults, I construct simple partial differential
equations that yield smooth unfaulting shifts for these samples. By solving these two different
kinds of equations simultaneously, I obtain unfaulting shifts for all samples in a seismic image,
and these shifts simultaneously move fault blocks and fault themselves to undo the faulting
in the seismic image.

Some limitations remain in the fault interpretation methods. One limitation is that I cur-
rently only estimate fault dip slips which are displacements in fault dip directions. In reality,
fault slips often consist of displacements in both dip and strike directions. The displacements
in strike directions, however, are more difficult to estimate because they tend to be parallel
to geologic layers and are difficult to resolve from a seismic image. Seismic stratigraphic fea-
tures (e.g. channels) extracted from seismic horizon surfaces might be used to estimate the
fault displacements in strike directions. Another limitation arises from the way I compute
unfaulting shifts for the samples away from faults using simple partial differential equations.
Although these simple equations can be solved efficiently and corresponding unfaulted re-
results appear reasonable, it might be possible and preferable to use a more geologically and
geomechanically correct way to compute the shifts for samples away from faults.

6.2 Unconformity interpretation

In Chapter 4, I propose a method to compute an unconformity likelihood image that
highlights both termination areas and the corresponding parallel unconformities. This uncon-
formity likelihood is defined as an attribute that evaluates the differences between two
seismic normal vector fields corresponding to two structure tensor fields constructed from a same seismic image using different smoothing filters. One structure-tensor field is constructed by applying a laterally structure-oriented smoothing filter and a vertical causal filter to each element of the outer products of seismic image gradients. The other one is constructed by applying a same laterally structure-oriented smoothing filter but a vertical anticausal filter.

Near the termination area of an unconformity, the reflector structures above and below the unconformity must be different. Therefore, the vertical causal filter, which computes locally averaged structures from above of the unconformity, yields a structure-tensor field that is different from the one constructed with the vertical anticausal filter, which computes locally averaged structures from below. The lateral structure-oriented smoothing filter extends the structure differences, which originate within the termination areas, to the corresponding parallel unconformities and correlative conformities. In doing this, the correlative conformity and parallel unconformity are assumed to be continuous throughout the seismic image. Therefore, if faults are apparent in a seismic image, we want to first undo the faulting in the seismic image so that seismic reflectors are continuous across faults, and then attempt to extract unconformities from the unfaulted image.

Using these lateral and vertical smoothing filters, the two constructed structure-tensors fields and the two corresponding vector fields should be different at both termination area and the corresponding parallel unconformity and correlative conformity. Therefore, the unconformity likelihoods, defined as the differences between the two vector fields, should be relatively high at both the angular unconformities and the corresponding parallel unconformities and correlative conformities. Unconformity surfaces, complete with termination areas and corresponding parallel unconformities or correlative conformities, can be extracted on the ridges of the unconformity likelihoods.

As an application, the extracted unconformity surfaces are used as constraints to estimate seismic normal vectors at unconformities with multiple-oriented seismic reflectors. A conventional structure-tensor method can correctly estimate the normal vectors (or slopes)
of the reflectors in conformable areas of a seismic image, but for an angular unconformity where multiple-oriented structures meet, this method yields smoothed vectors that represent averages of orientations across the unconformity. Using the extracted unconformity surfaces as constraints for the structure-tensor method, I obtain discontinuous seismic normal vectors at unconformities, which enable use to compute a flattened image with vertical gaps at unconformities.

In computing unconformity likelihoods, I use vertically causal and anticausal filters to obtain structure tensors that differ at unconformities because I assume unconformities are more horizontal than vertical. This assumption is often true but is not necessary. Therefore, for future work, unconformity likelihoods might be further improved by instead using causal and anticausal filters that smooth in directions orthogonal to unconformities.

6.3 Image flattening and horizon extraction

In Chapter 5, I propose two methods for computing seismic horizons from 3D seismic images. The first method extracts horizons one at a time, while the second one computes an entire volume of horizons at once by flattening the seismic image using a relative geologic time computed from seismic normal vectors. Both the methods enable an interpreter to interpret control points, scattered in the seismic image, to guide extracting single horizons and computing an entire volume of horizons. In both the methods, the interpreted control points are served as hard constraints, which are incorporated into the methods with simple constraint preconditioners in the conjugate gradient method used to compute single horizons and horizon volumes.

The automatic fault and unconformity interpretations discussed in Chapters 2, 3, and 4 facilitate automatic seismic image flattening, by correlating seismic reflectors across faults, providing accurately estimated seismic normal vectors at unconformities, and providing unconformities as constraints for flattening methods. However, manually interpreted constraints are still desirable for complicated areas in a seismic image where automatic methods might fail. The proposed methods provide an especially simple way for interpreters to im-
pose such constraints by interactively adding or moving control points in a seismic image, 
while efficiently updating a single seismic horizon or a complete horizon volume.

In the horizon volume method, I compute only vertical shifts to flatten a seismic image 
and then compute a volume of horizons. These vertical shifts, however, cannot flatten 
seismic images with nonvertical faults. Therefore, we might want to compute vector shifts, 
as discussed in Chapter 3, for unfaulting and flattening a seismic image, but might still use 
the same way to incorporate human interactions.

After flattening or generating an entire volume of horizons, one straightforward applica-
tion in the future is to use the horizon volume to guide interpolation of borehole data and 
obtain 3D images of subsurface properties, so that the property values conform to geologic 
structures followed by horizons. Moreover, horizontal slices of a flattened image reveal strati-
graphic features, such as channels, in 3D seismic images. Therefore, we might want to use 
those features to guide a sequence of 2D interpolations for horizontal slices in the flattened 
space, and then map the interpolated result back to the original space to obtain a structure-
and stratigraphic feature-guided interpolation.


——, 2013b, Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3D seismic images: Geophysics, 78, no. 2, O33–O43.


Luo, S., and D. Hale, 2013, Unfaulting and unfolding 3D seismic images: Geophysics, 78, no. 4, O45–O56.


——–, 2004, Relative geologic time (age) volumes—relating every seismic sample to a geologically reasonable horizon: The Leading Edge, 23, 928–932.


Wu, X., and D. Hale, 2015a, 3D seismic image processing for unconformities: Geophysics, 80, no. 2, IM35–IM44.

——–, 2015b, Horizon volumes with interpreted constraints: Geophysics, 80, no. 2, IM21–IM33.

——–, 2016, 3D seismic image processing for faults: GEOPHYSICS, 81, no. 2, IM1–IM11.


