Semiautomated fault interpretation based on seismic attributes

Bo Zhang[1], Yuancheng Liu[2], Michael Pelissier[3], and Nanne Hemstra[2]

Abstract

Three-dimensional fault interpretation is a time-consuming and tedious task. Huge efforts have been invested in attempts to accelerate this procedure. We present a novel workflow to perform semiautomated fault illumination that uses a discontinuity attribute as input and provides labeled fault surfaces as output. The procedure is modeled after a biometric algorithm to recognize capillary vein patterns in human fingers. First, a coherence or discontinuity volume is converted to binary form indicating possible fault locations. This binary volume is then skeletonized to produce a suite of fault sticks. Finally, the fault sticks are grouped to construct fault surfaces using a classic triangulation method. The processing in the first two steps is applied time slice by time slice, thereby minimizing the influence of staircase artifacts seen in discontinuity volumes. We illustrate this technique by applying it to a seismic volume acquired over the Netherlands Sector of the North Sea Basin and find that the proposed strategy can produce highly precise fault surfaces.

Introduction

Faults in the subsurface can act as barriers or efficient avenues for hydrocarbon migration and flow, and often form hydrocarbon traps. Identifying the fault system is one of first steps in seismic interpretation and a key component in developing exploration and development strategies. However, careful fault interpretation is a highly time-consuming task. Algorithms that facilitate fault interpretation fall into two categories. The first category deals with development and application of attributes that highlight fault locations. The algorithms in the second category are for generating fault surfaces from these attributes volumes.

Coherence/similarity (Bahorich and Farmer, 1995; Marfurt et al., 1998; Gersztenkorn and Marfurt, 1999; Randen et al., 2001), reflector dip (Marfurt, 2006), and curvature (Stewart and Wynn, 2000; Roberts, 2001; Al-Dossary and Marfurt, 2006) are the most popular seismic attributes routinely used to assist in fault interpretation. Unfortunately, attributes in their native form are not generally amenable to semiautomated fault system extraction. Rather, we need to apply additional edge-enhancement technology to these attributes to better illuminate faults and minimize human labor. There are a variety of image processing techniques which can enhance fault visualization and detection. Al-BinHassan and Marfurt (2003) employed the Hough transforms to enhance faults appearing on time slices. Aarre and Wallet (2011) generalized this workflow to three dimensions using an efficient add-drop algorithm. Barnes (2006) designed a filter to pass steeply dipping discontinuities which can serve as the first step in automating fault interpretation. Lavialle et al. (2006) have proposed a nonlinear filtering approach based on 3D GST analysis that denoises and preserves faults prior to automatic fault extraction. Image processing techniques applied to seismic attributes usually require a suitable window size. Larger window size not only smears the fault information but also increases the computational cost, whereas smaller window sizes introduces less smearing but are sensitive to noise.

Almost all automated fault extraction strategies need human intervention from time to time and include three main steps. First, the interpreter selects an appropriate fault-sensitive seismic attribute (e.g., coherence or reflector dip magnitude) to highlight the fault location. Next, the interpreter employs different technologies to transform the attribute volume into a fault likelihood/confidence volume. Finally, the interpreter generates a localized surface to fit a cloud of fault points. Randen et al. (2001) present a four-step workflow to automatically extract fault surface from an attribute cube. Unfortunately, this workflow does not handle X-pattern faults properly. Gibson et al. (2003) propose a two-step strategy to automatically detect the fault surface in 3D seismic data. The first step is to generate a confidence cube based on the coherence attribute. They then generate small patches and least-squares

[1]The University of Oklahoma, ConocoPhillips School of Geology and Geophysics, Norman, Oklahoma, USA. E-mail: bo.zhang-1@ou.edu.
[2]dGB Earth Sciences, Sugar Land, Texas, USA. E-mail: yuancheng.liu@dgbes.com; nanne.hemstra@dgbes.com.
[3]Formerly Marathon Oil Corporation; presently Roc Oil (Bohai) Company, Beijing, China. E-mail: mike.pelissier@rocoil.com.au.

Manuscript received by the Editor 19 May 2013; revised manuscript received 20 August 2013; published online 31 January 2014. This paper appears in *Interpretation*, Vol. 2, No. 1 (February 2014); p. SA11–SA19, 11 FIGS.
http://dx.doi.org/10.1190/INT-2013-0060.1.

fit those patches to generate a fault surface. In the Randen et al. (2001) and Gibson et al. (2003) workflows, the challenge lies in how to define a suitable threshold to generate the confidence volume as well as a proper window size to generate the fault surface. Silva et al. (2005) provide greater insight into the ant tracking algorithm proposed by Randen et al. (2001). They report that this strategy can reduce human interaction from 10 days to three days in their testing. Jacquemin and Mallet (2005) propose a method based on a cascade of two Hough transforms to automatically extract fault surfaces. Cohen et al. (2006) propose a workflow, which contains four steps to detect and extract fault surfaces in 3D volumes, resulting in a set of one-pixel-thick labeled fault surfaces. Kadlec et al. (2008) present a method to model faults surface using a growing surface strategy while Dorn et al. (2012) generated fault surfaces through azimuth scanning on horizontal slices, and dip scanning on vertical slices.

In this paper, we present a semiautomated strategy to extract fault surfaces from seismic attributes volumes that requires minimum human intervention. We start by introducing an edge-detection algorithm successfully used in the biometric field. We then use these edges to construct a fault system. Finally, we apply our algorithm to a seismic data volume acquired over the Netherlands Sector of the North Sea Basin.

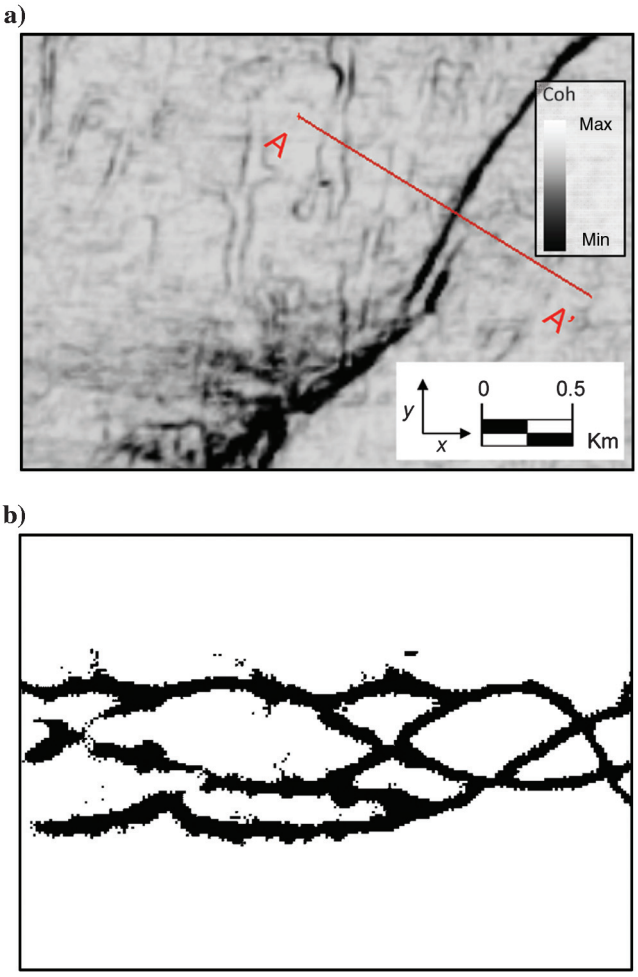

Figure 1. Patterns comparison between (a) seismic discontinuity attribute on time slice and (b) binarized vein plane (Modified from Miura et al., 2007). Those two objectives from different field show similar features in the plane.

Method

Coherence-like attributes typically highlight faults quite well on time/depth slices (Dorn et al., 2012) but usually exhibit a staircase behavior on the vertical sections. Based on this observation, we produce our fault sticks time slice by time slice prior to constructing the fault surfaces in the vertical direction.

Seismic attribute conditioning

The fault patterns shown on the time slices (Figure 1a) share similar characteristics with capillary vein images of fingers (Figure 1b) acquired using infrared light. Based on this observation, we borrow an effective method of extracting vein patterns (Miura et al., 2007) to recognize the fault elements on time slices. In their experiments, Miura et al. (2007) reduced the equal error rate (EER), which evaluates the mismatch ratios of personal identification, to 0.0009%. Whereas, the EER in other reported methods ranges from 0.2% to 4%. By calculating the local maximum curvature in cross sectional profiles of discontinuity attribute on time slices, the algorithm can extract the centerlines of possible fault locations. The output is a binarized volume where one indicates possible fault locations and zero the absence of faults.

Assume that P is an attribute slice and $P(x, y)$ is the value at grid (x, y). We define $P[\xi^{(j)}]$ as a cross sectional profile acquired from $P(x, y)$ along azimuth j, where $\xi^{(j)}$ is the position sequence number in the profile and (x, y) are, respectively, the index of inline and crossline number. For a given point of discontinuity attribute on time slice, our method checks the curvature $k[\xi^{(j)}]$ of cross sectional profiles $P[\xi^{(j)}]$ as a function of $\xi^{(j)}$ along azimuth j. The curvature $k[\xi^{(j)}]$ can be expressed as

$$k[\xi^{(j)}] = \frac{d^2P[\xi^{(j)}]/d[\xi^{(j)}]^2}{\{1 + \{dP[\xi^{(j)}]/d[\xi^{(j)}]\}^2\}^{\frac{3}{2}}}. \quad (1)$$

The shape of the attribute profile $P[\xi^{(j)}]$ is determined by the type of attribute. For example, coherence appears as a low coherence dent (Figure 2a) and exhibits negative curvature using equation 1. To simplify the following processing, if the attribute shows low values at the fault location, we reverse the sign of equation 1.

Note that the discontinuity attributes should theoretically reach minimum/maximum value at the fault location and increase/decrease abruptly (Figure 2b). We assume that the local maxima $k[\xi^{(j)}]$ in each profile $P[\xi^{(j)}]$ indicate the possible fault positions. Those points are defined as center positions $U^{(j)}(x, y)$. To determine

whether a center position $U^{(j)}(x,y)$ has the possibility to lie on the fault location, we compute scores $S[U^{(j)}(x,y)]$ (Figure 2c), defined as

$$S[U^{(j)}(x,y)] = k[U^{(j)}(x,y)] \times W[U^{(j)}(x,y)], \quad (2)$$

where $W[U^{(j)}(x,y)]$ is the local width of the profile where $k(\xi^{(j)})$ is positive (Figure 2b), and $k[U^{(j)}(x,y)]$ is valued directly from $k[\xi^{(j)}]$ from location mapping between (x, y) and $\xi^{(j)}$. The score parameter $S[U^{(j)}(x,y)]$ considers the width and changing rate of the attribute at the same time. If the score is large, the probability that there is a fault is also high. To obtain the fault pattern development along all azimuths in the entire time slice, the scores are accumulated and assigned to a capability plane (Figure 3), $V(x,y)$, which has the same size as the attribute time slice

$$V(x,y) = \sum_{j}^{J} S[U^{(j)}(x,y)], \quad (3)$$

where j the index of azimuth direction, j is the number of azimuth and set as eight in this paper, and (x, y) is the horizontal coordinate pair.

If $V(x,y)$ is large and has large values nearby, we consider this point lying on a fault system. Even if $V(x,y)$ is large but has small values nearby, a dot of noise is interpreted to occur at (x, y). To evaluate whether the capability slice encounters faults, we employ the strategy described by Miura et al. (2007),

$$C_0(x,y) = \min\{\max[V(x,y+1), V(x,y+2)], \max[V(x,y-1), V(x,y-2)]\}, \quad (4a)$$

$$C_{45}(x,y) = \min\{\max[V(x+1,y+1), V(x+2,y+2)], \max[V(x-1,y-1), V(x-2,y-2)]\}, \quad (4b)$$

$$C_{90}(x,y) = \min\{\max[V(x+1,y), V(x+2,y)], \max[V(x-1,y), V(x-2,y)]\}, \quad (4c)$$

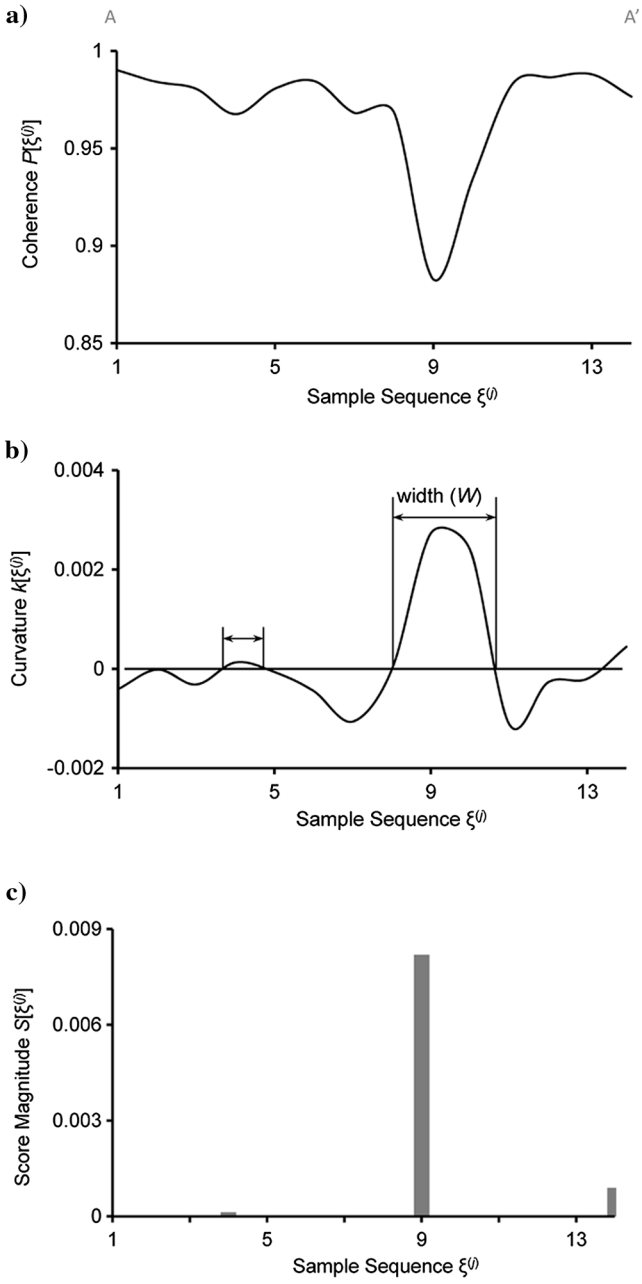

Figure 2. Diagrams showing the procedure of seismic attribute conditioning. The attributes value comes from the red line shown in Figure 1a. (a) Coherence serves as the input for the fault sensitive attribute. (b) The curvature computed from coherence attribute. (c) The score values used to output binary fault sticks.

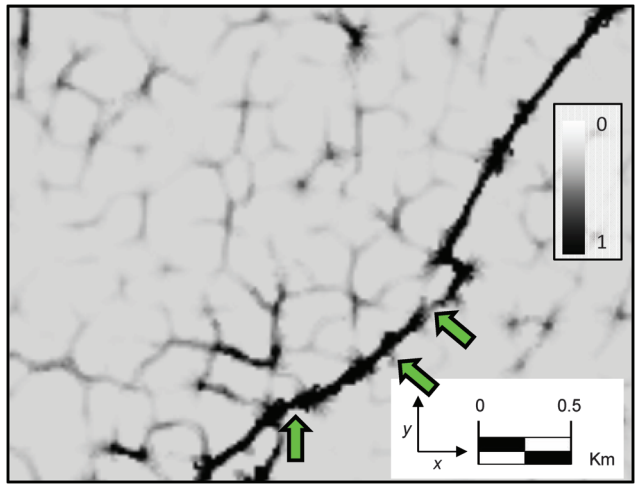

Figure 3. Capability time slice computed from the attribute slice shown in Figure 1a using the strategy of equation 3.

$$C_{135}(x,y) = \min\{\max[V(x+1,y-1), V(x+2,y-2)], \max[V(x-1,y+1), V(x-2,y+2)]\}. \quad (4d)$$

Figure 4 shows the confidence slices along different azimuths (0°, 45°, 90°, and 135°) after applying equation 4a–4d on the capability slice shown in Figure 3.

The final confidence estimate is given by

$$C(x,y) = \max[C_0(x,y), C_{45}(x,y), C_{90}(x,y), \text{and } C_{135}(x,y)]. \quad (5)$$

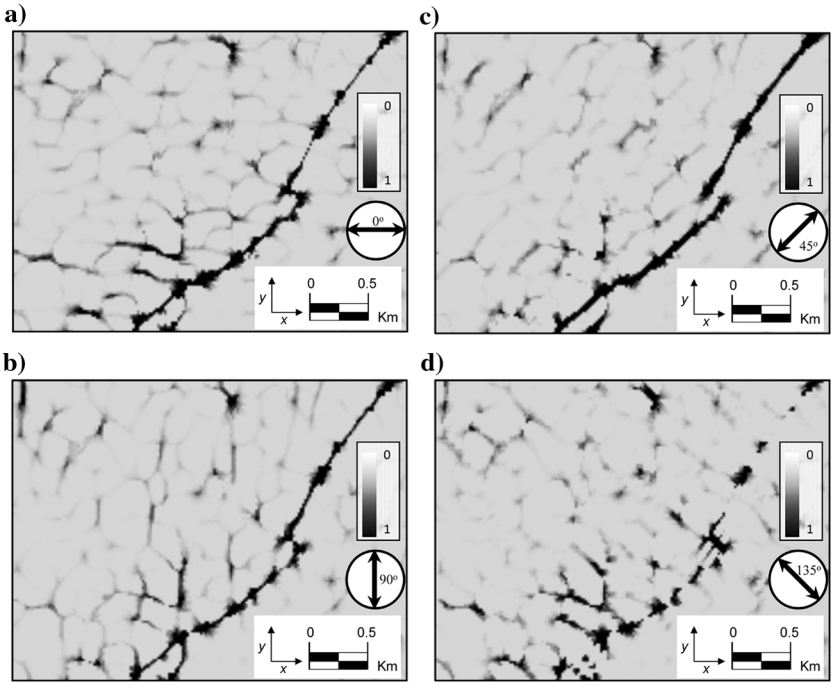

Figure 4. Confidence time slices encountering a fault at (a) 0°, (b) 90°, (c) 45°, and (d) 135° using equation 4a–4d applying on the capability time slice shown in Figure 3.

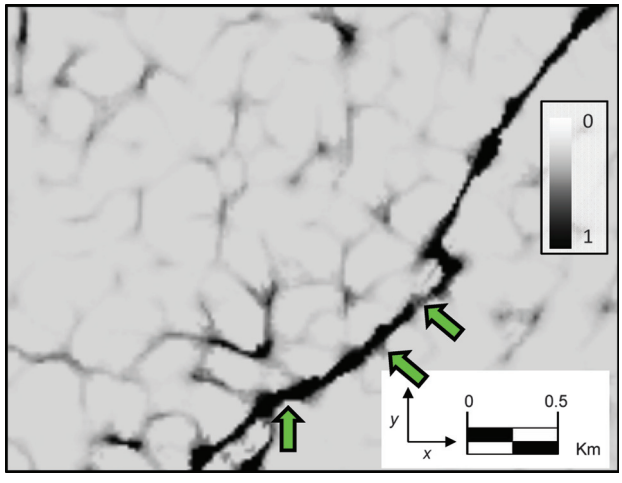

Figure 5. The final confidence estimated from Figure 4 using equation 5. We scale it to range between zero and one.

Note fault confidence attributes indicated by the green arrows in Figure 5 is more continuous compare to that of Figure 3. The improvement is critical in generating the binary slice.

The confidence slice is binarized according to a user-defined threshold (C_{thd} in Figure 7). Only those points with values greater than or equal to the threshold are set to one and considered as candidate points for the following processing and fault surface construction. All other points are treated as background with a value of zero (Figure 6a).

The above workflow is designed and set to highlight the faults and is applied to the whole seismic attribute cube time slice by time slice. The final result is a binarized cube where the points with value one indicate possible fault locations.

Thinning and connected component analysis

Thinning algorithms (e.g., Bag and Harit, 2011) applied to the binarized time slices can approximate the medial lines of the connected candidate points. The results are one-pixel thick lineaments that can also be used to separate different fault surfaces (Cohen et al., 2006). However, thinning may generate undesired bifurcation branches (indicated by blue arrows in Figure 6b) due to its sensitivity to noise and complex boundaries. Crossing fault surfaces also appear as bifurcated branches (indicated by the red arrows in Figure 6b) on the thinned slices. To determine whether a thinned stick has bifurcated branches, we examine the number of connected neighbor pixels (NCNP) for each pixel of current stick. A pixel is considered as the bifurcated point if its NCNP is greater than three and the stick has branches. We use the following criteria to preserve or trim the branches. If the length of the branches is much larger (e.g., three times for the examples shown in this paper) than the local width of the hypothesized binarized result at bifurcated point (e.g., the limb indicated by red arrow in Figure 6c), we assume the branches belong to some other fault surface. Otherwise, we simply trim the limbs and archive the maximum length of the current element (e.g., the limbs indicated by the blue arrows in Figure 6b). The length of the branches is determined by the number of pixel from bifurcated point till the end pixel of current limb (e.g., the length of branches indicated by the red arrow is 19 in Figure 6b). To determine the local width for binarized slice at the bifurcated point, we first draw a circle with a diameter of 1one pixel centered at the bifurcated point, and then increase the diameter until a pixel on circle

has value of zero (Figure 6a). At last, the local width is set as the diameter of the circle (e.g., the width labeled by red arrow is five in Figure 6a).

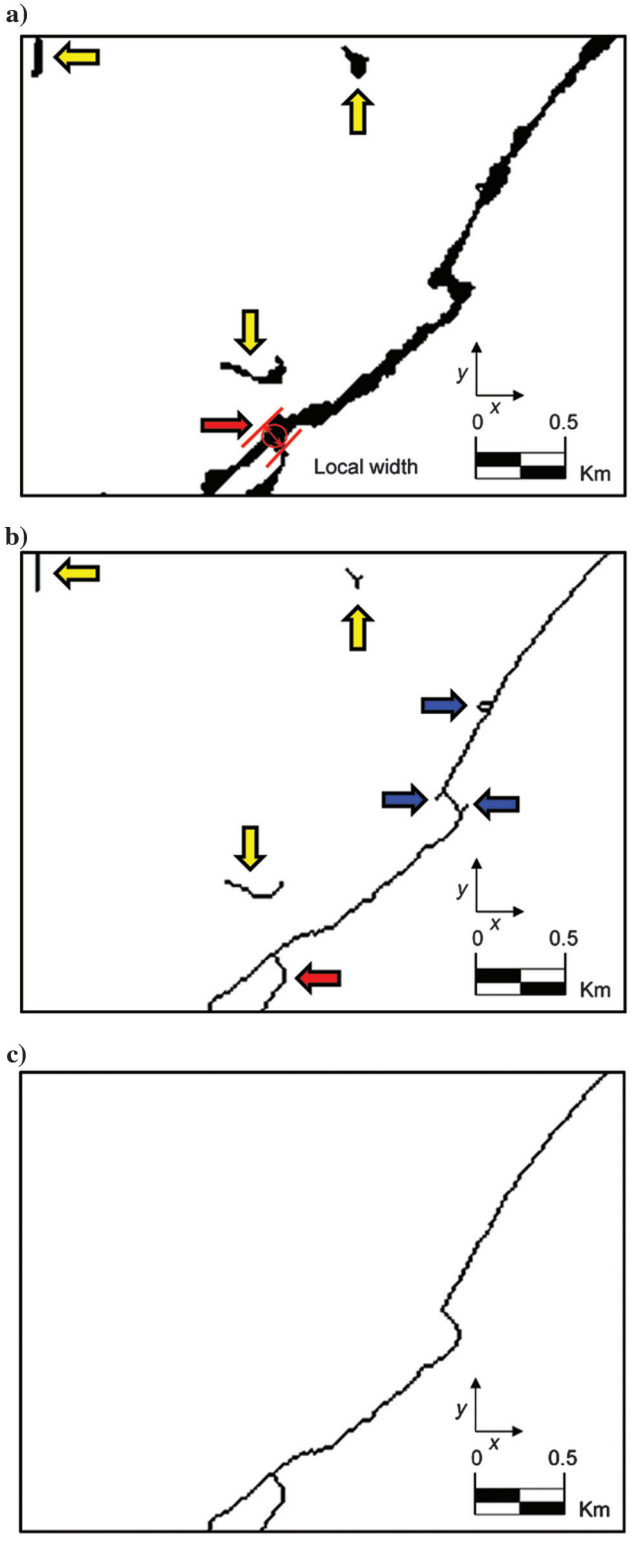

Figure 6. (a) Binarized slice after (b) thinning and (c) trimming processes. The binarization processing is applied on the time slice shown in Figure 5. The threshold value used in generating Figure 6a is 0.95.

Faults, stratigraphic edges, and acquisition footprint all give rise to elongated features on the trimmed time slice. To preserve the fault sticks only, we first use connected component analysis (e.g., Dillencourt et al., 1992) to label all the connected elements. We then only keep those components whose lengths are greater than or equal to a user-defined value ($L_{\min}$ in Figure 7). For example, the components indicated by yellow arrows in Figure 6b are deleted due to their limited length. This threshold also serves as the smallest length of the fault sticks we detect on each time slice. Figure 6c is the last output fault stick used for the following fault-generating surface.

Thinning, trimming, and component analysis are applied on the entire binarized cube time slice by time slice, resulting in a suite of linear fault elements on each slice ready for the final fault system construction. Channels often exhibit long linear elements on time slices and survive the initial fault sticks winnowing process. However, channels are stratigraphically limited and will in general only exhibit a few sticks vertically, which provides a means of rejecting them through the use of a vertical continuity threshold.

Interactive fault surface generation

The fault surface projected on the time slice is a suite of curves called fault sticks. Fault sticks on adjacent

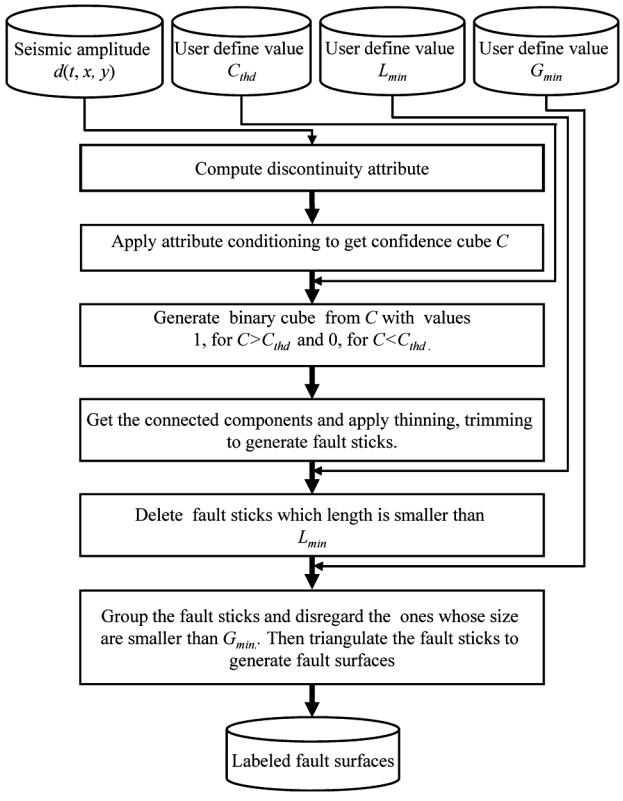

Figure 7. Flowchart showing the semiautomated fault interpretation based on seismic attributes. The whole procedure only requires three parameters which simplify the extraction processing.

time slices having similar size and shape are assumed to define the same geologic feature. Based on this assumption, we group the sticks by comparing their size and shape (e.g., Bribiesca and Aguilar, 2006). Starting with a given (source) stick, we search vertically ±4 samples over target sticks that share similar features with the source stick. Once a target stick is joined to the current fault surface, it is deleted from the sticks set and serves as the source stick to determine whether the next target stick is suitable for the current fault system. Once the stick grouping is done, we triangulate (e.g., Hartmann, 1998) the stick groups whose size is greater than or equal to a user-defined value (G_{min} in Figure 7) to generate a smooth fault surface. The suitable group size can reject not only the single noisy sticks, but also the channel-like long sticks. Interactive editing (e.g., merging) to ensure the fidelity of the extracted results is the final process in our workflow.

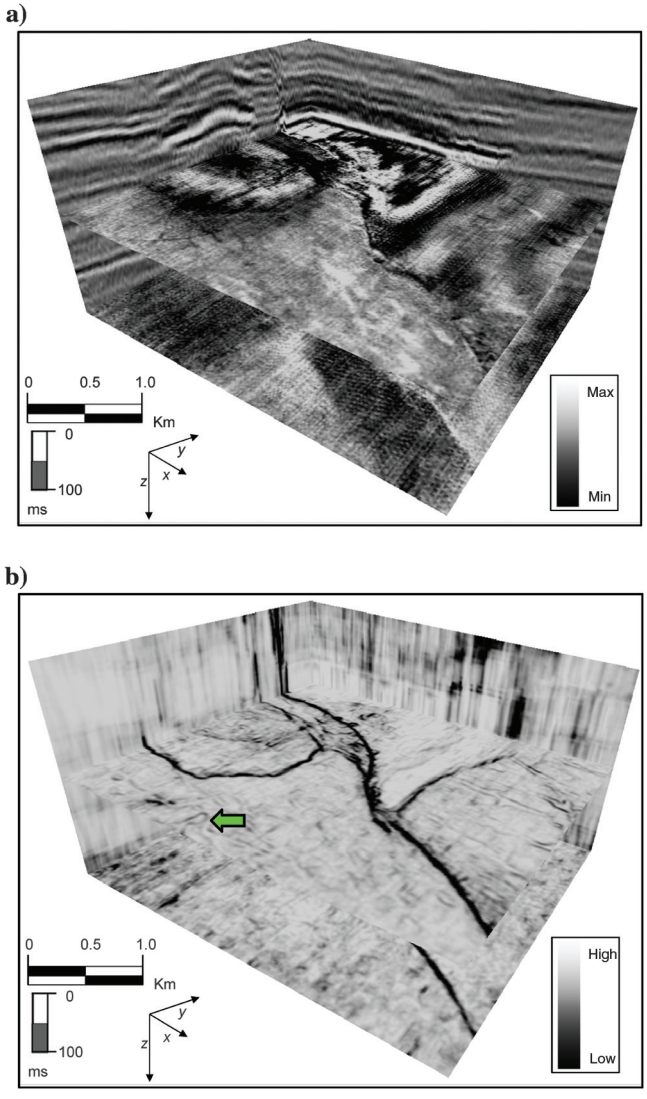

Figure 8. (a) Seismic amplitude and (b) coherence cube used for the algorithm testing.

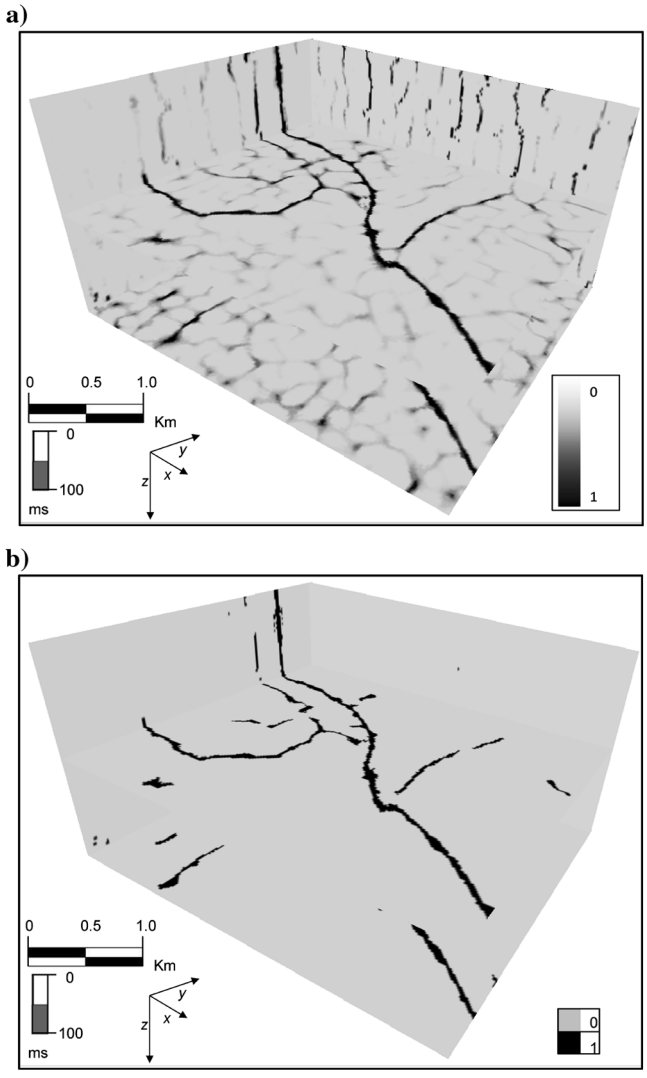

Figure 9. (a) Capability and (b) binarized cube computed from coherence attribute shown in Figure 8b.

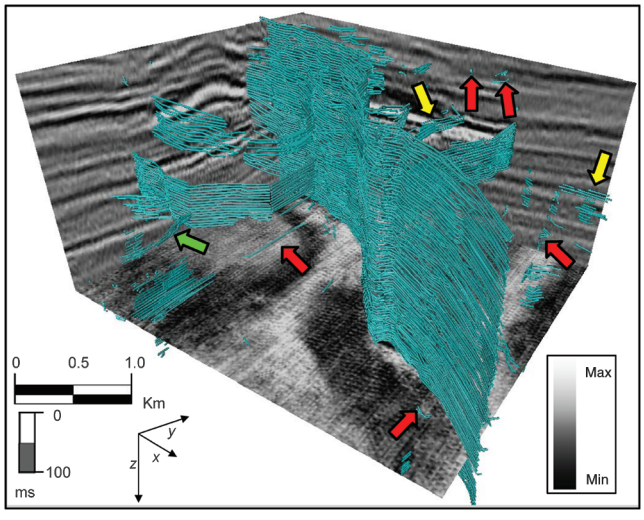

Figure 10. Three-dimensional view of trimmed fault sticks and original seismic data.

Figure 7 shows a workflow which summarizes fault surface extraction strategy in this paper. The input is seismic amplitude cube and outputs are labeled fault surfaces. We need three parameters to control the extraction procedure. The first parameter C_{thd} influences the generating of binary cube. The bigger the value of C_{thd}, the fewer pixels survive in the following processing. The second parameter, L_{min}, constrains the minimum length of fault sticks on horizontal slice, while the third parameter, G_{min}, controls fault surface size on vertical section.

Application

To demonstrate the capability and efficiency of our algorithm, we apply it to a subvolume of a seismic survey acquired in the Dutch portion of the North Sea Basin. Detailed mapping of the faults is critical to this survey because some of the faults may act as pathways for gas or fluids (Schroot and Schüttenhelm, 2003). The tested volume contains 250 × 200 traces and ranges from 300 to 700 ms with a sample interval of 4 ms.

Figure 8a shows the seismic cube with a major fault cutting data along one of the vertical faces. We choose coherence (Figure 8b) as the fault sensitive attribute. Note that the meandering channel indicated by the green arrow is shown in Figure 8b. We generate a capability cube C (Figure 9a) from coherence (Figure 8b) using the proposed conditioning strategy and scale it to range between zero and one. The binary cube is shown in Figure 9b with values one for $C > 0.95$ and zero for $C < 0.95$. Fault sticks generated from thinning and trimming are shown in the Figure 10. The previously described trimming successfully removes unwanted branches introduced by the thinning algorithm. Note that we still have unwanted sticks in Figure 10, such as noise sticks indicated by the red

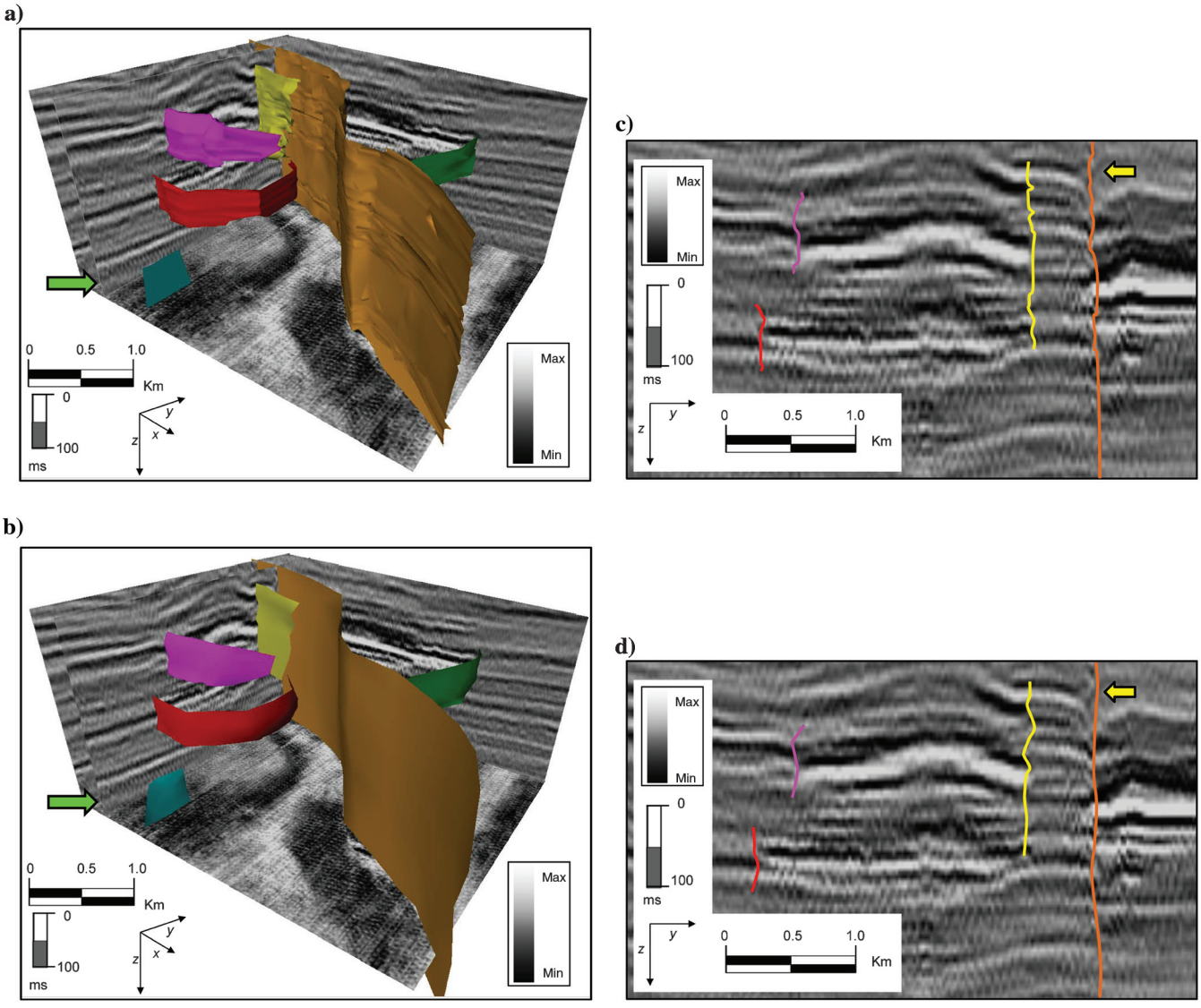

Figure 11. Visualization of the fault surfaces and original seismic data. Different color means different fault systems. (a) Extracted fault surfaces using the workflow shown in Figure 7. (b) Attribute-based manually interpreted fault surfaces. (c) Vertical section view of extracted fault surfaces. (d) Vertical section view of manually interpreted fault surfaces.

arrow and the channels sticks indicated by the green arrow. We choose a threshold value of 10 slices (40 ms) for the size of stick group to reject stratigraphic features. Figure 11a shows the final automated extracted fault surfaces labeled by different colors. Note that, by setting a threshold value of 10 (40 ms) for the size of stick group, the algorithm also deletes sticks belonging to two small faults indicated by the yellow arrows in Figure 10. Figure 11b is the manually interpreted fault surfaces based on coherence attribute shown in Figure 8b. We can see that there is good agreement between the automated and manually interpreted results. To better quality control the results, we respectively show vertical sections (indicated by green arrows in Figure 11a and 11b) with automated extracted and manually interpreted faults in Figure 11c and 11d. The yellow arrows in Figure 11c and 11d state our algorithm locates the fault surface better than that of manually interpreted results. Reducing time cost of human is the bright spot of our method. The whole procedure only requires about five minutes human intervention to generate all the fault surfaces. However, attribute-based manually interpretation needs about 20 minutes.

Discussion

The size of our subvolume is about 20 MB and whole computational cost is around 15 min on a single processor. The most time-consuming step is the generating of confidence cube, and it accounts for about 80% in our example. Through the parallelization of our algorithm, we can heavily speed up the whole extraction procedure. Parameter C_{thd} controls whether we can successfully generate desired faults surfaces. Because the cost of binary generating is negligible, our suggestion is that produces several binary cubes by setting different values of C_{thd} and uses the one that has connected pixels (pixels with value one) at the possible fault locations.

Conclusion

Understanding the fault system is a critical objective for any structural interpretation. The proposed algorithm and workflow facilitates this procedure by automatically generating fault surfaces from a discontinuity volume. There is no need for the tedious window size testing for attributes conditioning, and the whole procedure only needs three threshold values which simplify the fault conditioning process. The first threshold value is used for generating the binary cube. The second and third threshold values are, respectively, the lateral length of the fault stick and vertical size of the fault. The lateral length of the sticks controls the fault size apparent on time sections while the vertical size of the stick group determines the size of the fault on the vertical sections. Increasing the size of the stick group required to define a valid fault surface can reject noisy sticks, but may reject small faults. Note that the accuracy of our results is highly dependent on the quality of the seismic data. If the seismic data are so noisy that the coherence or other geometric attributes do not approximate faults, or if acquisition footprint is very strong, we do not recommend using an automated interpretation method.

Acknowledgments

The authors would like to thank TNO for providing the data and dGB Earth Sciences for the permission to publish this work. We also thank associated editor Arthus Barnes, reviewer Richard Dalley, reviewer Nasher M. AlBinHassanand, and the third anonymous reviewer. The final version of this paper benefitted tremendously from the comments and suggestions of Kurt J. Marfurt.

References

Aarre, V., and B. Wallet, 2011, A robust and compute-efficient variant of the Radon transform: Processing of the 31st Annual GCSSEPM Foundation Bob F. Perkins Research Conference, 550–586.

AlBinHassan, M. N., and K. J. Marfurt, 2003, Fault detection using Hough transforms: 73rd Annual International Meeting, SEG, Expanded Abstracts, 1719–1721.

Al-Dossary, S., and K. J. Marfurt, 2006, 3D volumetric multispectral estimates of reflector curvature and rotation: Geophysics, **71**, no. 5, P41–P51, doi: 10.1190/1.2242449.

Bag, S., and G. Harit, 2011, Skeletonizing character images using a modified medial axis-based strategy: International Journal of Pattern Recognition and Artificial Intelligence, **25**, 1035–1054.

Bahorich, M., and S. Farmer, 1995, 3-D seismic discontinuity for faults and stratigraphic features, The coherence cube: 65th Annual International Meeting, SEG, Expanded Abstracts, 93–96.

Barnes, A. E., 2006, A filter to improve seismic discontinuity data for fault interpretation: Geophysics, **71**, no. 3, P1–P4, doi: 10.1190/1.2195988.

Bribiesca, E., and W. Aguilar, 2006, A measure of shape dissimilarity for 3D curves: International Journal of Contemporary Mathematical Sciences, **1**, 727–751.

Cohen, I., N. Coult, and A. Vassiliou, 2006, Detection and extraction of fault surfaces in 3D seismic data: Geophysics, **71**, no. 4, P21–P27, doi: 10.1190/1.2215357.

Dillencourt, M., H. Samet, and M. Tamminen, 1992, A general approach to connected component labeling for arbitrary image representations: Journal of the Association for Computing Machinery, **39**, 253–280.

Dorn, G., B. Kadlec, and P. Murtha, 2012, Imaging faults in 3D seismic volumes: 82nd Annual International Meeting, SEG, Expanded Abstracts, 1–5.

Gersztenkorn, A., and K. J. Marfurt, 1999, Eigenstructure-based coherence computations as an aid to 3D structural and stratigraphic mapping: Geophysics, **64**, 1468–1479, doi: 10.1190/1.1444651.

Gibson, D., M. Spann, and J. Turner, 2003, Automatic fault detection for 3D seismic data: Proceedings of Digital

Image-Techniques and Applications Conference, Expanded Abstracts, **1**, 821–830.
Hartmann, E., 1998, A marching method for the triangulation of surfaces: The Visual Computer, **14**, 95–108.
Jacquemin, P., and J. L. Mallet, 2005, Automatic faults extraction using double Hough transform: 75th Annual International Meeting, SEG, Expanded Abstracts, 755–758.
Kadlec, B. J., G. A. Dorn, H. M. Tufo, and D. A. Yuen, 2008, Interactive 3-D computation of fault surfaces using level sets: Visual Geosciences, **13**, 133–138, doi: 10.1007/s10069-008-0016-9.
Lavialle, O., S. Pop, C. Germain, M. Donias, S. Guillon, N. Keskes, and Y. Berthoumieu, 2006, Seismic fault preserving diffusion: Journal of Applied Geophysics, **61**, 132–141.
Marfurt, K. J., 2006, Robust estimates of 3D reflector dip and azimuth: Geophysics, **71**, no. 4, P29–P40, doi: 10.1190/1.2213049.
Marfurt, K. J., R. L. Kirlin, S. H. Farmer, and M. S. Bahorich, 1998, 3D seismic attributes using a running window semblance-based algorithm: Geophysics, **63**, 1150–1165, doi: 10.1190/1.1444415.
Miura, N., A. Nagasaka, and T. Miyatake, 2007, Extraction of finger-vein patterns using maximum curvature points in image profiles: IEICE TRANSACTIONS on Information and Systems, **E90-D**, 1185–1194.
Randen, T., S. I. Pedersen, and L. Sønnelan, 2001, Automatic extraction of fault surfaces from three-dimensional seismic data: 71st Annual International Meeting, SEG, Expanded Abstracts, 551–554.
Roberts, A., 2001, Curvature attributes and their application to 3D interpretation horizons: First Break, **19**, 85–100.
Schroot, B. M., and R. T. E. Schüttenhelm, 2003, Expressions of shallow gas in the Netherlands North Sea: Netherlands Journal of Geosciences, **82**, 91–105.
Silva, C., C. Marcolino, and F. Lima, 2005, Automatic fault extraction using ant tracking algorithm in the Marlim South Field, Campos Basin: 75th Annual International Meeting, SEG, Expanded Abstracts, 857–860.
Stewart, S. A., and T. J. Wynn, 2000, Mapping spatial variation in rock properties in relationship to scale-dependent structure using spectral curvature: Geology, **28**, 691–694.

Bo Zhang received a B.S. (2006) in geophysics from China University of Petroleum, and an M.S. (2009) in geophysics from the Institute of Geology and Geophysics, Chinese Academy of Sciences. He is currently a doctoral student at the University of Oklahoma working on his thesis titled "Long offset seismic analysis for resources plays."

Biographies and photographs of the other authors are not available.