

# Semiautomatic fault-surface generation and interpretation using topological metrics

Yihuai Lou<sup>1</sup>, Bo Zhang<sup>1</sup>, Pan Yong<sup>2</sup>, Huijing Fang<sup>3</sup>, Yijiang Zhang<sup>4</sup>, and Danping Cao<sup>5</sup>

## ABSTRACT

Seismic fault surfaces are compulsory input for structure modeling that unravels the structural deformation history of the subsurface. Seismic fault attributes provide geoscientists with alternative images of faults. However, seismic fault attributes only highlight possible fault locations and do not directly provide fault surfaces that are required inputs for structural modeling. Interpreters construct seismic fault surfaces using interpreted seismic fault sticks on vertical seismic slices. Interpreting fault sticks on hundreds of seismic slices is time consuming. We have semiautomatically constructed fault surfaces by simulating the procedure of manual seismic fault interpretation. Our algorithm consists of three main steps:

(1) obtaining fault sticks in the inline, crossline, and time slices; (2) grouping the fault sticks according to the connectivity and mutual exclusion (topology) between the fault sticks on the inline, crossline, and time slices; and (3) generating the fault surface patches by merging the fault sticks time slice by time slice through the topology analysis. Our algorithm contains one optional step: manually merging the fault patches if needed. We test our algorithm on open access seismic data and our workflow accurately generates fault surfaces for most faults including conjugate faults in the seismic data. Considering that it usually helps to weight the estimation according to the quality of the computed fault attribute, the algorithm computes fault parameters such as fault dip and strike using weighted principal component analysis.

## INTRODUCTION

Fault identification is critical for identifying potential drilling hazards and for understanding the orientation and intensity of potential natural fractures. Researchers have developed seismic fault attributes to assist seismic fault interpretations in 3D seismic surveys. Research related to highlighting the possible seismic fault location can be classified into two categories. The first category generates seismic fault (or discontinuity) attributes to highlight the possible fault locations. The earliest discontinuity seismic attributes are based on the crosscorrelation of seismic traces. Unfortunately, crosscorrelation is sensitive to noise. However, the crosscorrelation provides a starting point for the later discontinuity

algorithms. [Marfurt et al. \(1998\)](#) extend Bahorich and Farmer's crosscorrelation approach using multiple and analytical seismic traces, referred to as semblance-based coherence. The analytical seismic trace is robust to noise contained in the seismic data ([Barnes, 2007](#)). One disadvantage to the method of [Marfurt et al. \(1998\)](#) is that it is sensitive to lateral variation in seismic amplitude. [Gersztenkorn and Marfurt \(1999\)](#) compute the coherence (referred to as eigenstructure-based coherence) by analyzing the eigenstructure of seismic traces that is not sensitive to the lateral amplitude variation of seismic traces. [Marfurt \(2006\)](#) further improves the accuracy of coherence by applying a multiple Kuwahara window search to the computed coherence attribute. Although the seismic coherence attribute successfully highlights the possible fault loca-

Manuscript received by the Editor 19 January 2020; revised manuscript received 21 December 2020; published ahead of production 21 January 2021; published online 11 March 2021.

<sup>1</sup>The University of Alabama, Department of Geological Science, Tuscaloosa, Alabama 35487, USA. E-mail: ylou6@crimson.ua.edu; bzhang33@ua.edu (corresponding author).

<sup>2</sup>Bohai Oilfield Research Institute, Tianjin Branch of CNOOC Ltd, Tianjing 300452, China. E-mail: panyong@cnooc.com.cn.

<sup>3</sup>The University of Alabama, Department of Geological Science, Tuscaloosa, Alabama 35487, USA and China University of Petroleum (Beijing), College of Geoscience, Changping-Qu, Beijing 102249, China. E-mail: hfang11@ua.edu.

<sup>4</sup>The University of Alabama, Department of Geological Science, Tuscaloosa, Alabama 35487, USA and Chengdu University of Technology, College of Geophysics, Chengdu 610059, China. E-mail: yzhang322@ua.edu.

<sup>5</sup>China University of Petroleum (East China), School of Geoscience, Qingdao 266580, China. E-mail: caodp@upc.edu.cn.

© 2021 Society of Exploration Geophysicists. All rights reserved.

tion, undesired sequence and staircase artifacts are also common in the computed coherence attributes. Assuming that the fault surface at each time sample can be represented by a “small” plane, Lou et al. (2019) generate the fault attribute without staircase artifacts and undesired sequence artifacts by using a local fault model.

The second category aims to mitigate the artifacts contained in seismic fault attributes and to improve the continuity of fault attributes on vertical and horizontal slices. Pedersen et al. (2002, 2003) propose the ant-tracking algorithm to enhance fault features along paths of “artificial ants.” AlBinHassan and Marfurt (2003) assume that the fault and fractures can be represented by local straight lines and then enhance the continuity of faults and fractures by converting the pixels (samples) from the space domain to the polar domain by using the Hough transform. The disadvantage of this approach is that it cannot handle strong background noise and preserve the true length of faults and fractures at the same time. To enhance the lineament of faults and fractures without obviously changing the length of faults and fractures, other coherence enhancing algorithms are based on smoothing the coherence along the orientation of the fault plane (Neff et al., 2000; Cohen et al., 2006; Hale, 2009; Qi and Castagna, 2013). Obtaining an accurate estimation for the orientation of the fault plane is the main challenge for the smoothing algorithms. It is extremely hard to only include the faults and fractures pixels (samples) in the process of orientation estimation. Qi et al. (2017, 2018) improve the accuracy of orientation estimation by applying principal component analysis (PCA) on skeletonized fault images. A minor disadvantage of the method proposed by Qi et al. (2017, 2018) is that it cannot produce an accurate orientation estimation for the conjugate faults.

Researchers have tried to automatically generate seismic fault surfaces that can be used in the structural models using seismic fault attributes. The fault surface and the corresponding fault parameters are very useful for geologists and can be used for evaluating the fault seal properties. Zhang et al. (2014) generate fault surfaces by applying the vein pattern recognition algorithm to coherence attributes. The vein pattern recognition algorithm can automatically generate fault sticks but needs interpreters manually grouping the generated fault sticks that belong to the same fault surface. Wu and Hale (2016) generate fault surfaces by analyzing the linked structure between potential fault pixels observed using seismic attributes. Defining a proper window size to interpolate the missing neighbor fault pixels is the main challenge for this method. A small window size may leave a “hole” in the constructed fault surface, and a large window size may include pixels that do not belong to the fault surface. Wu and Fomel (2018) propose the optimal surface voting algorithm to generate fault surfaces and calculate fault parameters such as fault dip, fault strike, and fault throw. One minor disadvantage of that method is that it cannot accurately handle fault surfaces with opposing dip angles at different locations.

In this paper, we propose a new workflow to semiautomatically generate fault surfaces from seismic attribute data. This paper is organized as follows. The “Nomenclature” section introduces terminology used throughout this paper. The “Data description” section describes the two seismic surveys used in this paper. The “Method” part consists of four sections: workflow, generating fault sticks, grouping fault sticks, and generating fault surfaces. We illustrate our workflow step by step by applying it on the poststack seismic survey of Kerry, acquired offshore New Zealand. We further demonstrate the effectiveness of our workflow by applying it to the

Netherlands F3 survey. The “Discussion” section illustrates the steps in setting proper values for the parameters used in the proposed workflow. The final section is the “Conclusion.”

## NOMENCLATURE

In the following, we introduce a few key concepts and representations that will be used to describe the construction of fault surfaces from seismic attributes. We define “manual fault interpretation” as the picking of fault sticks on user-defined 2D vertical seismic sections (e.g., using the Petrel E&P software platform from Schlumberger). The fault pixel, which represents a point of the surface, is a pixel that is located on or nearby a potential fault surface. The fault stick, which represents the fault trace, is a line or curve on inline, crossline, or time slices. Fault pixels that belong to the same fault stick can be grouped into a connected set of pixels. Thus, the fault stick consists of “a swarm of” interconnected fault pixels on a 2D inline, crossline, or time slice. A fault surface patch, which is a part of the fault surface, is a relatively small surface that consists of several interconnected fault sticks that belong to the same fault surface. A fault surface, which represents the fault plane in the subsurface, consists of a swarm of interconnected fault pixels in the subsurface. The fault surface only has one fault pixel (sample) along the direction perpendicular to the fault plane at the fault location. The interpretation software generates a fault surface by applying a certain interpolation algorithm to the manually picked fault sticks. In this paper, the fault surface construction is based on fault pixels that represent points in the subsurface.

## DATA DESCRIPTION

We use two open-access seismic surveys (Kerry and F3) to demonstrate the effectiveness of the proposed method (dGB Earth Sciences, 1987). Both 3D marine seismic surveys were provided by New Zealand Crown Minerals. We use the Kerry seismic survey to illustrate the proposed method in detail. We only show the final extracted fault surfaces of seismic survey F3 due to the limited space of this article. The Kerry seismic survey used in this paper consists of 200 inlines and 600 crosslines. The F3 seismic survey used in this paper consists of 400 inlines and 800 crosslines. The bin size of both surveys is  $25 \times 25$  m in the inline and crossline directions, and both seismic surveys were sampled at 4 ms. The two-way traveltime of the Kerry and F3 seismic surveys ranges from 0.1 to 1.8 s and 0.5 to 1.8 s, respectively.

## METHOD: WORKFLOW DESCRIPTION

Our method consists of three steps, and it is based on analyzing the connectivity and mutual exclusion between fault sticks on 2D slices (Figure 1). Generating fault sticks is the first step in our workflow. The algorithm begins with generating “initial” fault sticks on the inline, crossline, and time slices from the seismic fault attributes using a user-defined threshold. The algorithm then computes the azimuth of the initial fault sticks on the time slice and the dip of the initial fault sticks on vertical slices by integrating a Kuwahara window and the weighted PCA (WPCA). We obtain refined fault sticks by analyzing the azimuth and dip of the initial fault sticks. Then, the refined fault sticks are grouped to the same group if they are connected with each other directly or indirectly. Finally, we produce fault surface patches from the grouped fault sticks by analyz-

ing their topological relationships. The workflow consists of an optional step: manually merging the fault surface patches if needed.

## METHOD: GENERATING FAULT STICKS USING SEISMIC FAULT ATTRIBUTES

Generating fault sticks using seismic fault attributes is the first step of fault surface construction. Interpreters manually draw fault sticks on 2D vertical and horizontal seismic slices. Similarly, we generate faults sticks on inline, crossline, and time slices using our proposed method. Figure 2 shows the seismic fault attributes obtained using the method proposed by Lou et al. (2019) overlaid on seismic slices. Lou et al. (2019) calculate the seismic fault attribute to highlight fault locations using a local fault model. The step of fault stick generation begins with a binarization process applied to seismic fault attribute according to a user-defined threshold  $f_{\min}$ . The pixels on the 2D slices with the zero and one values are regarded as nonfault and fault pixels after binarization, respectively. The interconnected fault pixels are treated as an unanalyzed fault stick. We next generate the initial fault sticks by analyzing each unanalyzed fault stick individually. We compute the orientation of the initial fault sticks by integrating a Kuwahara window searching method and WPCA. The orientations of the initial fault sticks on vertical slices and time slices are the fault dip and fault azimuth, respectively. We finally obtain refined fault sticks by analyzing the orientation of the initial fault sticks.

### Generating the initial fault sticks

Figure 3a and 3b shows a representative cropped 2D seismic and seismic fault attribute slice, respectively. Figure 3c shows the corresponding binarized result. The potential fault pixels (the black color in Figure 3c) are regarded as the unanalyzed fault pixels. Figure 3d shows one swarm of interconnected unanalyzed fault pixels. Note that we can divide the unanalyzed fault stick into several fault sticks and each fault stick may belong to a different fault surface. The red arrow in Figure 3d indicates a hole within the unanalyzed fault stick. To facilitate the following thinning processing, we fill the hole by assigning pixels at the holes with the value of one. Figure 3e and 3f shows the refined unanalyzed fault stick and thinned unanalyzed fault stick using the thinning workflow proposed by Zhang et al. (2014), respectively. There are two types of fault pixels after thinning: (1) those with two neighboring pixels and (2) those having three or more neighboring pixels. We name the pixels with three or more neighbor pixels as furcated pixels. The red dot in Figure 3g indicates the location of the furcated pixels. We divide the unanalyzed fault stick into several independent parts at the furcated pixels, and each part is treated as an independent fault stick. The divided independent fault sticks are called the initial fault sticks. At the same time, we delete the thinned initial fault stick whose pixel number is smaller than a user-defined value  $l_{\min}$ . The threshold  $l_{\min}$  is treated as the smallest fault stick that we want to extract within the seismic survey. The blue arrow in Figure 3g indicates one thinned initial fault stick, which we ignore in the following processing. To stabilize the following orientation computation, we “expand” the divided fault sticks (Figure 3g) to their originally unthinned size (Figure 3h) by performing the morphological dilation (Gonzalez and Woods, 1992). We name the expanded fault sticks (Figure 3h) as the initial fault sticks.

### Calculating the orientation of the initial fault sticks

We compute the orientation of the initial fault sticks using WPCA analysis, and we determine the orientation of each fault pixel using Kuwahara window searching (Marfurt, 2006). We use model coordinates  $(x, y)$  to describe the location of each fault pixel located within the 2D slice. Considering that the pixels with a high fault probability value should have more weight in the fault orientation

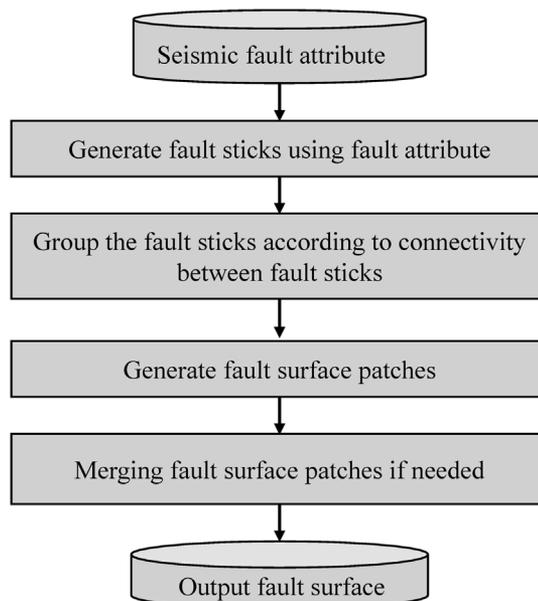


Figure 1. The workflow of simulating the process of manual seismic fault interpretation.

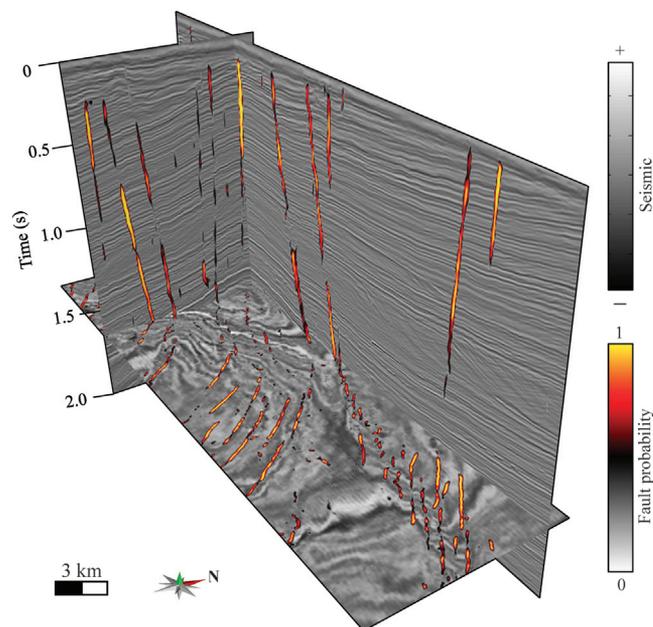


Figure 2. Chair diagram showing the fault attribute overlaid on the corresponding seismic data, which is acquired from the New Zealand Kerry survey.

computation, we use WPCA to compute the orientation of each fault pixel. To compute the orientation of the analyzed fault pixel, we first extract the fault pixels centered at the analyzed pixel. The window used to extract the fault pixels is a rectangular window, and the size of the window is  $l_{\min}$  by  $l_{\min}$  pixels. If there are  $N$  fault pixels centered at the analyzed pixel, then the coordinate vector of the fault pixels can be expressed as  $\mathbf{x} = (x_1, x_2, \dots, x_N)$  and  $\mathbf{y} = (y_1, y_2, \dots, y_N)$ . The seismic fault attribute values

for the fault pixels are expressed as  $\mathbf{s} = (s_1, s_2, \dots, s_N)$ . WPCA implements a weighted PCA procedure to each fault pixel within the analysis window by adding a weight to the covariance matrix  $\mathbf{C}_w$  (Fan et al., 2011) using

$$\mathbf{C}_w = \frac{1}{N-1} \begin{bmatrix} \text{cov}_w(\mathbf{x}, \mathbf{x}) & \text{cov}_w(\mathbf{x}, \mathbf{y}) \\ \text{cov}_w(\mathbf{y}, \mathbf{x}) & \text{cov}_w(\mathbf{y}, \mathbf{y}) \end{bmatrix}, \quad (1)$$

where  $N$  is the number of fault pixels centered at the analyzed pixel and the covariance elements  $\text{cov}_w(\mathbf{x}, \mathbf{x})$ ,  $\text{cov}_w(\mathbf{x}, \mathbf{y})$ ,  $\text{cov}_w(\mathbf{y}, \mathbf{x})$ , and  $\text{cov}_w(\mathbf{y}, \mathbf{y})$  are defined as follows:

$$\text{cov}_w(\mathbf{x}, \mathbf{x}) = \frac{\sum_{i=1}^N s_i (x_i - \bar{x})(x_i - \bar{x})}{N-1}, \quad (2a)$$

$$\text{cov}_w(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^N s_i (x_i - \bar{x})(y_i - \bar{y})}{N-1}, \quad (2b)$$

$$\text{cov}_w(\mathbf{y}, \mathbf{x}) = \frac{\sum_{i=1}^N s_i (y_i - \bar{y})(x_i - \bar{x})}{N-1}, \quad (2c)$$

$$\text{cov}_w(\mathbf{y}, \mathbf{y}) = \frac{\sum_{i=1}^N s_i (y_i - \bar{y})(y_i - \bar{y})}{N-1}, \quad (2d)$$

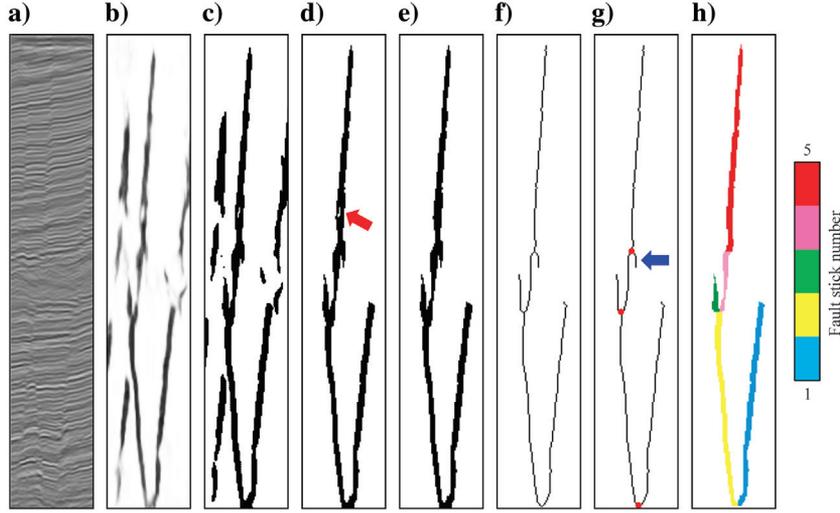


Figure 3. The process of generating initial fault sticks. (a) The seismic section, (b) the fault attribute section, (c) the binarized result of fault attribute, (d) the unanalyzed fault stick, (e) the refined unanalyzed fault stick, (f) the thinned unanalyzed fault stick, (g) the thinned unanalyzed fault stick with furcated pixels indicated by the red dots, and (h) the initial fault sticks.

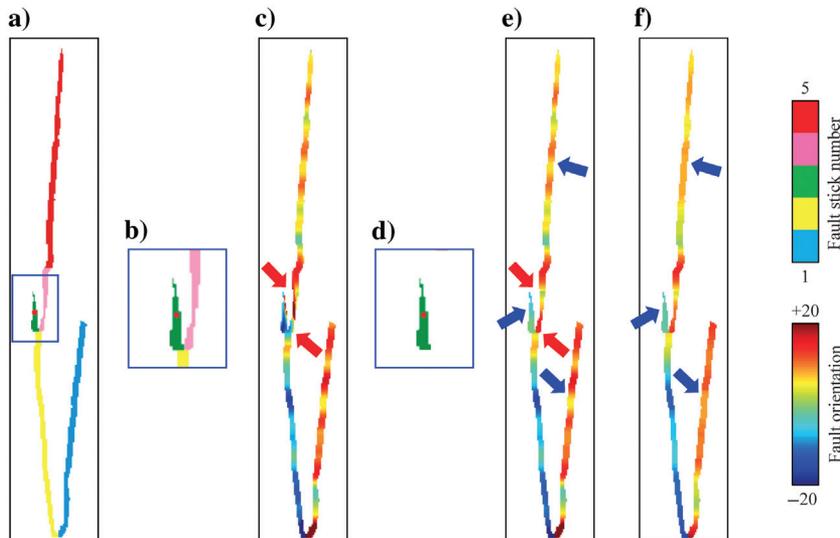


Figure 4. The calculated fault orientation using the traditional and our proposed method. (a) The analysis point (the red dot) and the corresponding analysis window (the blue rectangle) that are centered at the analysis point. (b) The traditional method uses all of the samples that are located within the analysis window. (c) The calculated fault orientation using the traditional method. (d) The proposed analysis window. (e) The calculated fault orientation using the proposed method. (f) The refined fault orientation after applying the Kuwahara searching method to the calculated fault orientation shown in (e).

where  $\bar{x}$  and  $\bar{y}$  are the average value of  $\mathbf{x}$  and  $\mathbf{y}$ , respectively. We obtain the eigenvalue and eigenvector of the covariance matrix shown in equation 1 by applying the eigendecomposition

$$\mathbf{C}_w = \lambda_u \mathbf{u} \mathbf{u}^T + \lambda_v \mathbf{v} \mathbf{v}^T = [\mathbf{u} \ \mathbf{v}] \begin{bmatrix} \lambda_u & 0 \\ 0 & \lambda_v \end{bmatrix} \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \end{bmatrix}, \quad (3)$$

where  $\lambda_u$  and  $\lambda_v$  are the eigenvalues satisfied by  $\lambda_u \geq \lambda_v \geq 0$  and  $\mathbf{u}$  and  $\mathbf{v}$  are the corresponding normalized eigenvectors. According to the PCA theory, the eigenvector  $\mathbf{v}$  is perpendicular to the fault stick at the analysis pixel. Therefore, the fault orientation  $p$  and confidence  $q$  of the analysis pixel are defined as

$$p = \arctan\left(\frac{v_y}{v_x}\right), \quad (4a)$$

$$q = \frac{\lambda_u}{\lambda_u + \lambda_v}, \quad (4b)$$

where  $v_x$  and  $v_y$  are the elements of eigenvector  $\mathbf{v}$  in the  $x$ - and  $y$ -directions, respectively; we have  $0 \leq q \leq 1$ . The unit of  $p$  is degrees.

We obtain the orientation  $\mathbf{p} = (p_1, p_2, \dots, p_N)$  and the corresponding confidence  $\mathbf{q} = (q_1, q_2, \dots, q_N)$  for all the pixels of the fault stick by applying the method illustrated in equations 1–4. We refine the calculated orientation of fault sticks by applying the Kuwahara window searching to the confidence vector  $\mathbf{q}$ . The window size for the Kuwahara window search is  $l_{\min}$  by  $l_{\min}$  pixels.

Figure 4 shows the strategy of the traditional analysis window and the proposed analysis window. The red dot in Figure 4a is one representative analysis fault pixel. The blue rectangle in Figure 4b is the analysis window used for fault orientation computation if we only consider whether a pixel is a fault pixel (the traditional analysis window is centered at the analysis pixel). The blue rectangle in Figure 4d is the analysis window (the most coherent window) used for fault orientation computation by considering to which fault stick the analysis pixel belongs. Note that the traditional analysis window in Figure 4b includes fault pixels belonging to two faults whose orientation differs with each other. As a result, we may obtain an inaccurate orientation value if the analysis window includes bifurcated fault pixels. However, our new analysis window (Figure 4d) only contains fault pixels belonging to the same fault stick. Thus, we will obtain a more accurate orientation estimation. Figure 4c and 4e shows the computed orientation using the traditional analysis window and our new strategy, respectively. The red arrows in Figure 4c and 4e indicate the notably improved orientation estimation. Figure 4f shows the refined fault orientation after applying the Kuwahara window searching to the calculated orientation shown in Figure 4e. Note that the refined fault orientation (indicated by the blue arrows) in Figure 4f is more uniform and accurate than the calculated fault orientation in Figure 4e.

### Generating refined fault sticks

We generate the initial fault sticks (Figure 3f) by analyzing the furcation properties of thinned fault pixels. However, we notice that we need to further analyze the separated fault sticks. Figure 5a shows one initial vertical fault stick with no furcated fault pixels. However, the upper and lower parts of the fault stick shown in Figure 5a have opposing orientations. Although the orientation of fault surfaces may vary with location, it is rare that one vertical fault stick has opposite orientations. Figure 6a shows five initial fault sticks on one vertical slice that belongs to two fault surfaces (conjugate faults). Note that the initial fault sticks number four and five can be merged into one fault stick. Thus, we propose to refine the initial fault sticks by analyzing the fault stick orientation. There are two operations in the refining process: merging and separating. We merge two initial fault sticks that are interconnected on the binaried slice if the orientation difference between the two initial fault sticks is smaller than a user-defined threshold  $\theta_{\text{thrd}}$ . In this paper,  $\theta_{\text{thrd}}$  is set as  $5^\circ$ . We separate the initial fault sticks if the orientation difference between two nearby faults pixels is larger than  $\theta_{\text{thrd}}$ . Figures 5b and 6b show the refined fault stick orientation. The white arrows in Figure 5b indicate the separating locations of the initial fault stick. The white arrow in Figure 6b indicates the merging location. Figures 5c and 6c show the refined fault sticks by using the orientation value shown in Figures 5b and 6b, respectively.

We apply our proposed fault stick refining step to every vertical and horizontal seismic slice. The refined fault sticks on the vertical (inline and crossline) and horizontal (time slice) slices are called vertical fault sticks and horizontal fault sticks, respectively. We give a unique fault stick number to each refined fault stick on the vertical

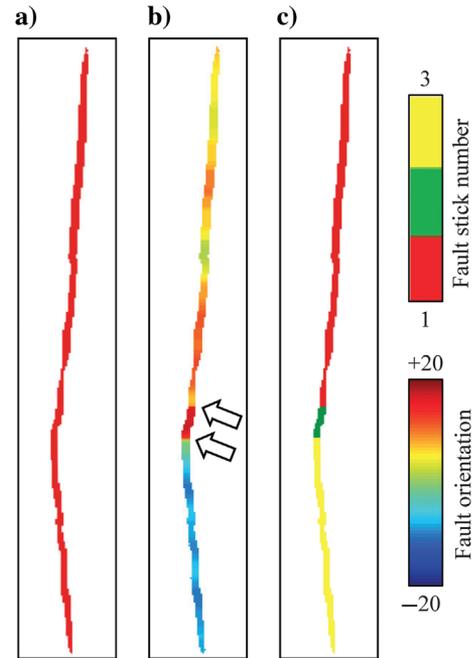


Figure 5. The representative example of fault stick separating. (a) The initial fault stick, (b) the refined fault orientation, and (c) the refined fault stick.

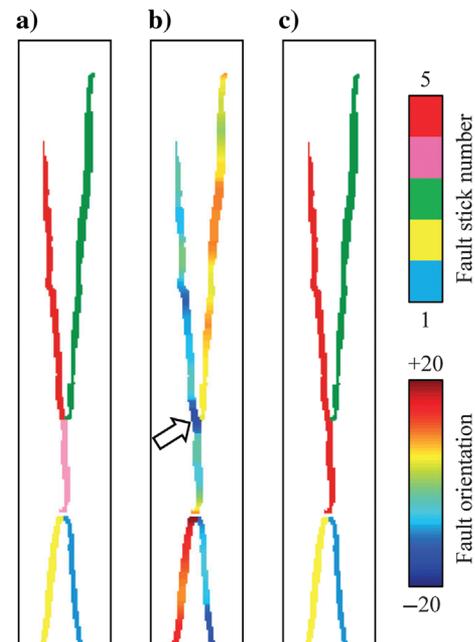


Figure 6. The representative example of fault stick merging. (a) The initial fault stick, (b) the refined fault orientation, and (c) the refined fault stick.

and time slices. Figure 7a shows the refined fault azimuth on the time slice and the fault dip on the vertical slices. Figure 7b shows the refined fault sticks on the time and vertical slices. As a result, each time sample of seismic traces has three fault stick numbers if the time sample is located at the fault location.

**METHOD: GROUPING FAULT STICKS**

To facilitate the following fault surface construction, we propose grouping the fault sticks (Figure 8) into different groups prior to the fault surface construction. The fault sticks within the same group should belong to the same surface if there is no intersection between two or more fault surfaces. All of the refined fault sticks form the stick bank used for grouping. The longest horizontal fault stick within the stick bank is regarded as the “center” stick for generating a group of fault sticks. The workflow then collects all vertical fault sticks that are connected to the center stick. We form new center fault sticks by merging all of the horizontal fault sticks connected with the collected vertical fault sticks. The grouping process iteratively collects vertical fault sticks and merges horizontal fault sticks until there are no more fault sticks connected to any of the fault sticks in the current group. We then assign a group number to the grouped fault sticks and exclude the fault sticks that already are merged with the current group. We next locate the longest horizontal fault stick within the stick bank and repeat the above procedure. The grouping continues until there are no more fault sticks within the fault stick bank.

The red fault stick shown in Figure 9 is the longest horizontal fault stick within our seismic survey, and it functions as the center stick in the first grouping round. We then extract all vertical fault sticks (the red sticks in Figure 10a) that are interconnected with the selected center stick (the blue stick in Figure 10a). The blue stick in Figure 10a is the 3D display of the red stick in Figure 9. The red and blue sticks together in Figure 10a function as the new center sticks (the blue sticks in Figure 10b), and we continue extracting sticks (the red sticks in Figure 10b) from the stick bank that are intercon-

nected with the current center sticks. The new extracted sticks iteratively become center sticks, and we iteratively extract sticks from the stick bank that are interconnected with the current center sticks. Figure 10c shows all of the extracted fault sticks of stick group number one and those fault sticks interconnected with each other directly or indirectly. Note that fault sticks within group number one belong to a set of conjugate faults.

**METHOD: FAULT SURFACE GENERATION**

The fault sticks within the same group number are interconnected to each other directly or indirectly. The interconnected fault sticks

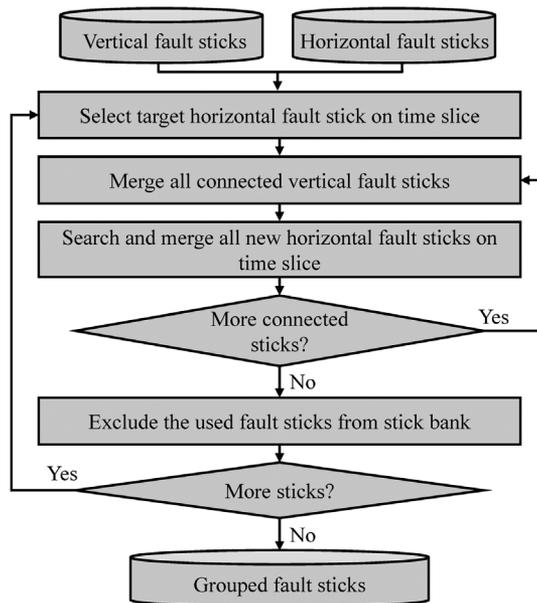


Figure 8. Fault stick grouping workflow.

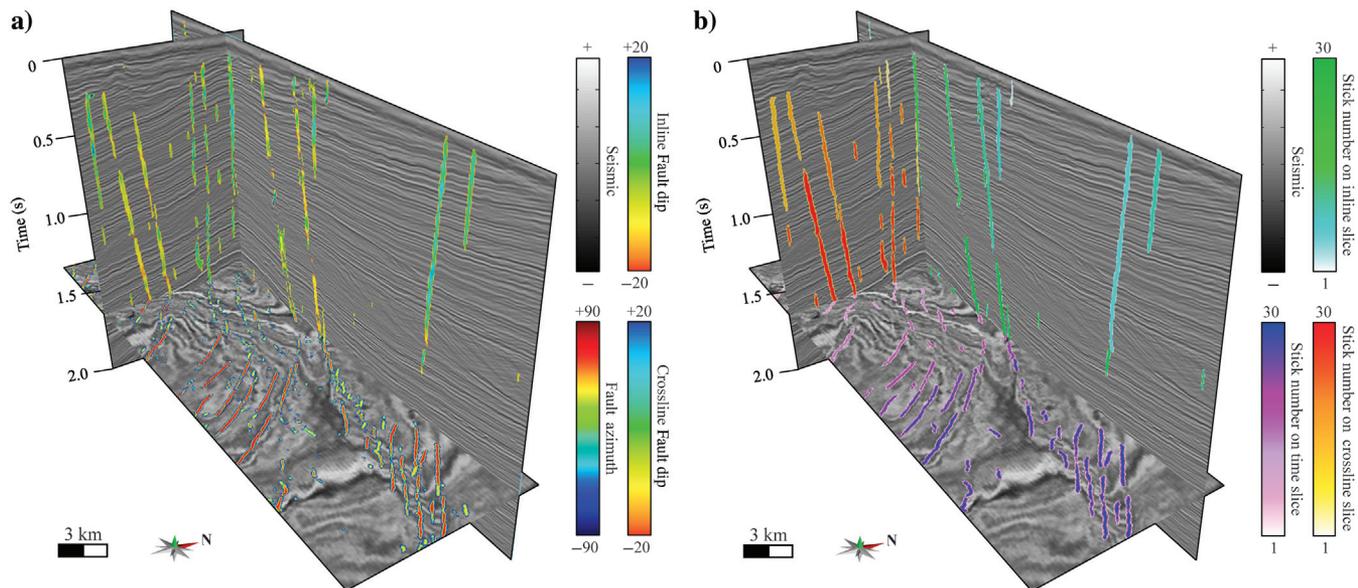


Figure 7. Chair diagram showing (a) the refined fault dip and azimuth and (b) the generated vertical and horizontal fault sticks overlaid on the seismic survey.

may belong to one or more fault surfaces. Our fault surface construction procedure is implemented on each grouped fault stick, and the surface construction procedure consists of three steps. We first generate fault surface patches by analyzing the topological relationship between fault sticks on vertical slices. We then automatically merge the fault patches by analyzing the topological relationships between the fault sticks of the fault surface patches. Our workflow also contains an optional step: manually merging the fault patches. The last step is computing the fault dip and azimuth using 3D WPCA analysis.

### Generating fault surface patches

Fault surface patch construction is based on analyzing the topological relationship of fault sticks on vertical slices. Figure 11a shows an inline slice with three vertical fault sticks. It is obvious that the black and green fault sticks belong to different fault surfaces. Note that we have two unconnected fault pixels when we extract an array of fault pixels along the time slice (the dashed blue line in Figure 11a). Thus, we can determine whether the fault sticks on the vertical slices belong to different fault surfaces by examining the connectivity within the array of fault pixels extracted on different time indices of the vertical slices. The black and green sticks shown in Figure 11a are called the “mutual-exclusion vertical sticks” because they belong to different fault surfaces. The fault surface patch construction merges the horizontal fault sticks time slice by time slice (Figure 11b) if there is no mutual exclusion between the vertical fault sticks that contain the fault pixels of horizontal fault sticks. We begin the merging with a user-defined time slice that contains the longest horizontal sticks, and the merging procedure is simultaneously implemented on the time slices above and below the user-defined time slice. The merging procedure continues until we have mutual-exclusion vertical sticks on the corresponding vertical slices (indicated by the red arrow in Figure 11c).

Figure 12 illustrates the workflow for the fault surface patch generating and labeling strategies. Figures 13, 14, and 15 illustrate three of the most common scenarios for the surface patch generating and labeling strategies.

Figure 13 shows an illustration of one of the labeling scenarios. The red horizontal stick on time slice T1 is regarded as the starting fault stick for merging. Note that two blue horizontal sticks on time

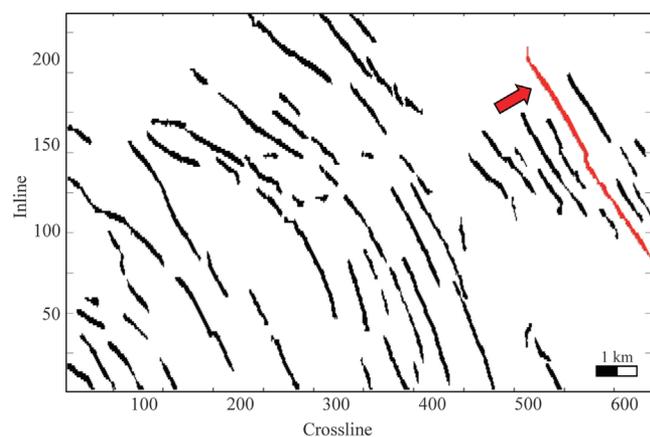


Figure 9. The horizontal fault sticks on a representative time slice. The red arrow indicates the selected horizontal fault stick.

slice T2 are indirectly connected to the red stick of time slice T1 through the vertical sticks. Note that there is no mutual exclusion between the vertical sticks that are connected with the two horizontal blue sticks. We assume that the two blue sticks belong to the same fault surface A, although they are two individual sticks on time slice T2. Note that the yellow fault stick on time slice T3 is indirectly connected to both blue sticks on time slice T2. Thus, the yellow fault stick on time slice T3 belongs to fault surface patch A.

Figure 14 shows another scenario for the horizontal sticks on neighboring time slices. The red stick on time slice T1 is the starting stick for merging. There are two sticks on time slice T2 that are indirectly connected with the red stick on time slice T1 through

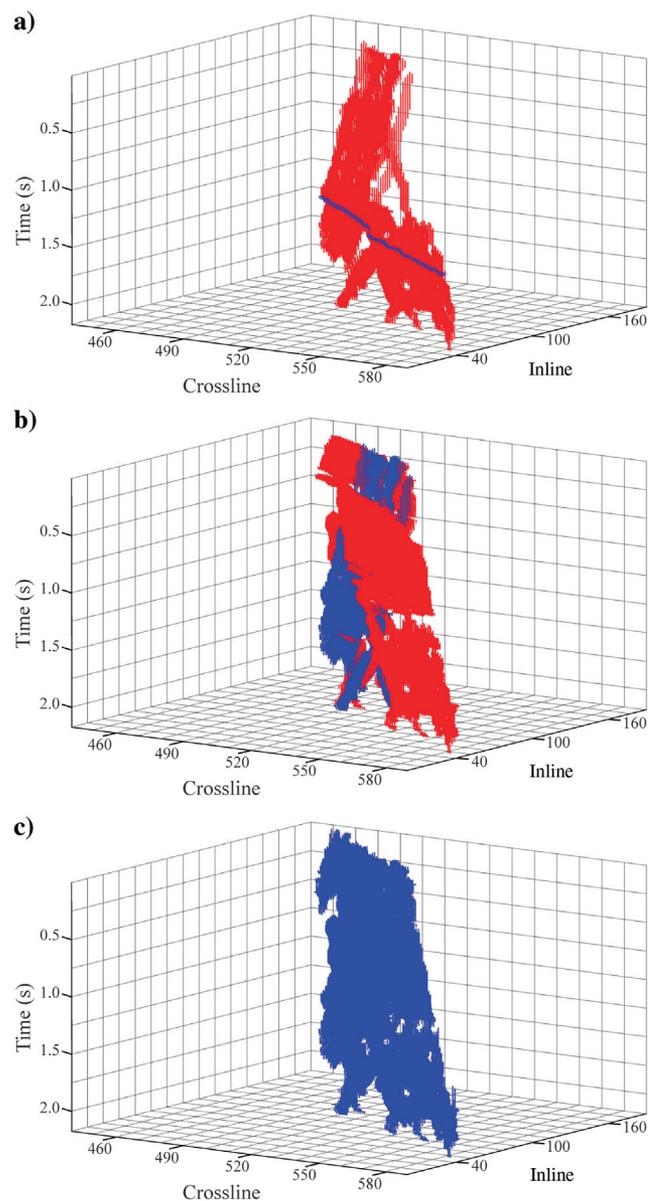


Figure 10. The representative example of the fault stick grouping operation. (a) The center stick (blue) and the grouped vertical fault sticks (red) in the first grouping round. (b) The new center sticks (blue) and the grouped horizontal fault sticks (red). (c) All of the extracted fault sticks.

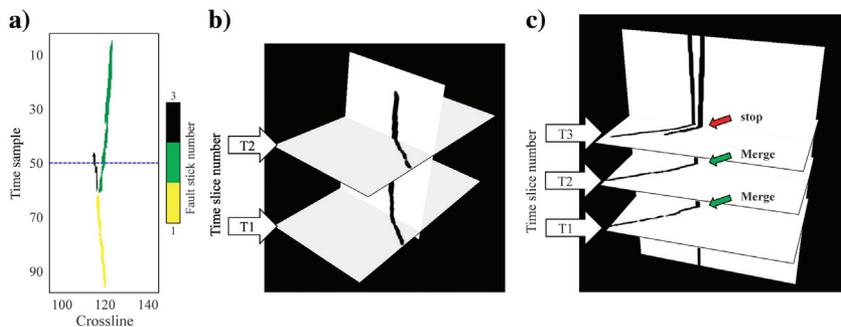


Figure 11. (a) The representative inline slice with three vertical fault sticks. The black and green sticks are the mutual-exclusion vertical sticks. The representative examples of the fault surface patch construction operations (b) without and (c) with the mutual-exclusion vertical sticks.

the vertical fault sticks. Note that we have mutual exclusions between the vertical sticks that are connected to the two individual horizontal sticks on time slice T2. We name the fault surface that contains the red stick in Figure 14 as fault surface patch A. Then, the fault sticks on time slice T2 are given as A-1 and A-2, meaning that they are connected with fault surface patch A directly or indirectly. The fault surface patches A-1 and A-2 are treated as new fault surfaces in the following merging analysis, and fault surface patch A terminates at time slice T2. Note that there is no mutual exclusion between the vertical fault sticks of fault surfaces A and A-1. In addition, there is no mutual exclusion between the vertical fault sticks of fault surfaces A and A-2. Thus, we

should merge fault surface A with one of them (surfaces A-1 and A-2). The fault surface patch called A-1 (or A-2) means that surface A-1 (or A-2) is connected with surface A. However, fault surface patch A-1 is mutually exclusive with fault surface patch A-2. In this manner, the following merging processing can quickly detect the candidate fault surface patches that need to be merged. For example, the merging algorithm would first try to merge surface patch A-1 (or A-2) if fault surface patch A is the analysis surface patch. Fault surface patch A cannot merge with surface patch A-2 if surface patch A is already merged with surface patch A-1, and vice versa.

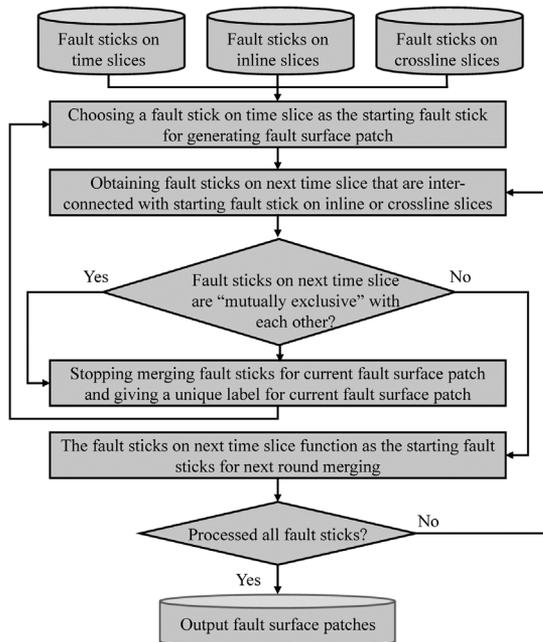


Figure 12. The workflow for the fault surface patch generation and labeling strategies.

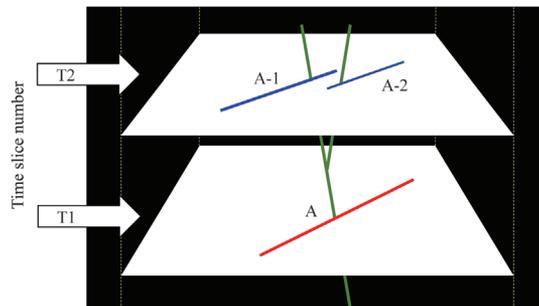


Figure 14. An illustration of the labeling strategy with the mutual-exclusion vertical sticks.

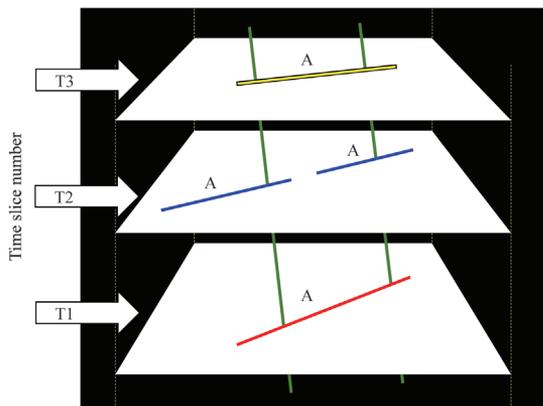


Figure 13. An illustration of the labeling strategy without the mutual-exclusion vertical sticks.

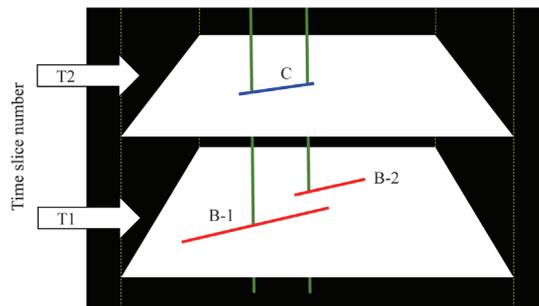


Figure 15. An illustration of the labeling strategy that shows the example of stopping the merging fault sticks on the current time slice and starting a new fault surface merging on the next time slice.

Figure 15 shows one more scenario for the horizontal sticks on neighboring time slices. The red stick B-1 is treated as the starting stick for merging. The blue stick on time slice T2 is indirectly connected to stick B-1 on time slice T1. The blue stick on time slice T2 is also connected to stick B-2 on time slice T1. However, sticks B-1 and B-2 on time slice T1 are mutually exclusive to each other. We propose to stop merging all of the fault sticks on time slices that are above time slice T1 and to start a new fault surface merging (fault surface C).

Figure 16a–16c shows fault sticks of different fault surfaces on representative time slices after the topological analysis using the process illustrated in Figures 13–15. Figure 17 shows fault sticks of different fault surface patches on a representative inline slice after topological analysis. Note that the inline slice shown in Figure 17 may have three fault surfaces. It is obvious that fault surface patches A-1 and A-2 in Figure 17 belong to different fault surfaces. However, it is possible that fault surface patches A and A-1 belong to a same fault surface. Thus, we propose merging the fault surface patches by analyzing the topological relationship between or among different fault surface patches.

**Merging fault surface patches**

We define two criteria for the merging of fault surface patches. The first criterion is that the analyzed fault surface patch can only merge with its neighboring fault surface patches if it shares vertical fault sticks with the analysis fault surface patch. Ideally, the fault surface patch should not be allowed to merge with any neighboring fault surface patches that have mutual-exclusion vertical sticks with the analyzed fault surface patch. However, considering the accuracy of the input fault attribute, we allow the fault surface patch merging neighboring fault surface patches if the percentage of exclusive fault sticks is less than a user-defined threshold  $s_{min}$  (e.g., 5% in this study). The second criterion is that the merging begins with the neighboring fault surface patch that shares the most vertical fault sticks with the analyzed fault surface patch. The analyzed fault surface patch only merges with one neighboring fault surface patch each time, and the merged fault surface patch functions as the new analyzed fault surface patch in the following merging. The merging process continues until the algorithm cannot merge any two fault surface patches.

Figure 18 shows the automatically merged fault surface patches. The red, cyan, part of the blue (indicated by the red arrow in Figure 18), and the yellow fault surface patches belong to the same fault surface. The remaining part of the blue and the green fault surface patches belong to another fault surface. However, we must crop that part of the blue fault surface patch indicated by the red arrow prior to the merging between the blue and green fault surface patches. The inaccurate input fault attribute is responsible for the inaccurate merging indicated by the red arrow in Figure 18. Figure 19a shows the fault sticks of a conjugate fault on representative time slices. Figure 19b shows the interpreted fault on a representative vertical slice. In this figure, time slice T2 only has one fault stick; time slices T1 and T3 have two fault sticks only if there is enough of a horizon gap (the upper and lower time slices) between the two fault surfaces. Considering that there is an inaccurate merging for the blue fault surface patch, we propose a step of manual

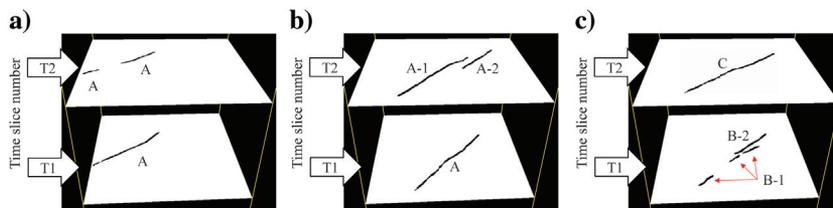


Figure 16. Fault sticks of different fault surfaces on representative time slices after the topological analysis. (a) Three fault sticks that belong to the same fault surface. The fault sticks on time slice T2 are unconnected with each other. However, both are connected with the fault stick on time slice T1 through vertical fault sticks. (b) Three fault sticks that belong to two fault surfaces. Both fault sticks on time slice T2 are connected with the fault stick on time slice T1 through vertical fault sticks. However, fault sticks A-1 and A-2 are mutually exclusive with each other on time slice T2. (c) Fault sticks that are belong to different fault surfaces.

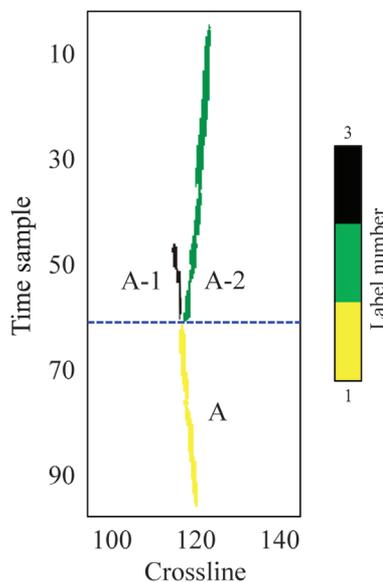


Figure 17. Fault sticks of different fault surface patches on representative inline slice after the topological analysis.

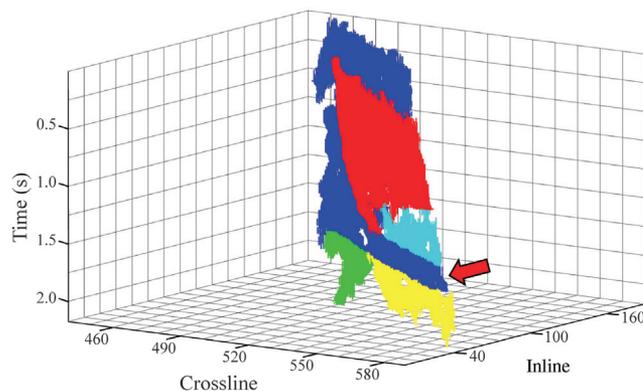


Figure 18. The automatic merged fault surface patches.

fault surface patch merging if needed. We first merge the fault surface patches (red, cyan, and yellow) that are accurately produced in the automatic merging step. Considering that the output fault surface should be a continuous surface in the subsurface, the fault surface merges the vertical fault sticks that are directly connected to the red, cyan, and yellow fault surface patches. We delete the fault pixels of the blue fault surface patch that have been used by another fault surface. Figure 20a shows the result after we manually merge the fault patches.

The final step of fault surface generation is interpolating a surface passing through the fault sticks. Figure 20b shows the final generated fault surfaces. We successfully generate two fault surfaces from

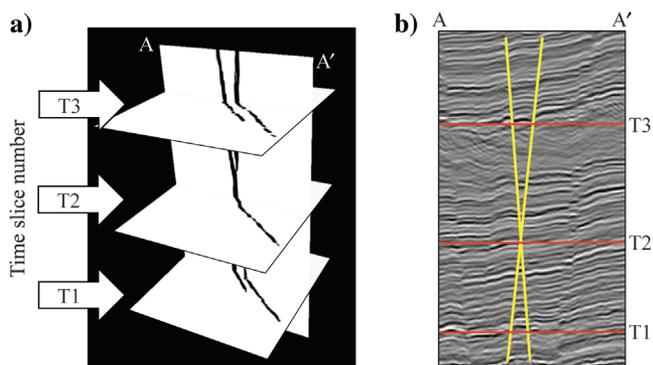


Figure 19. The fault sticks of the conjugate faults on the representative time and vertical slices. (a) The fault sticks and (b) the interpreted conjugate faults on the vertical seismic slice.

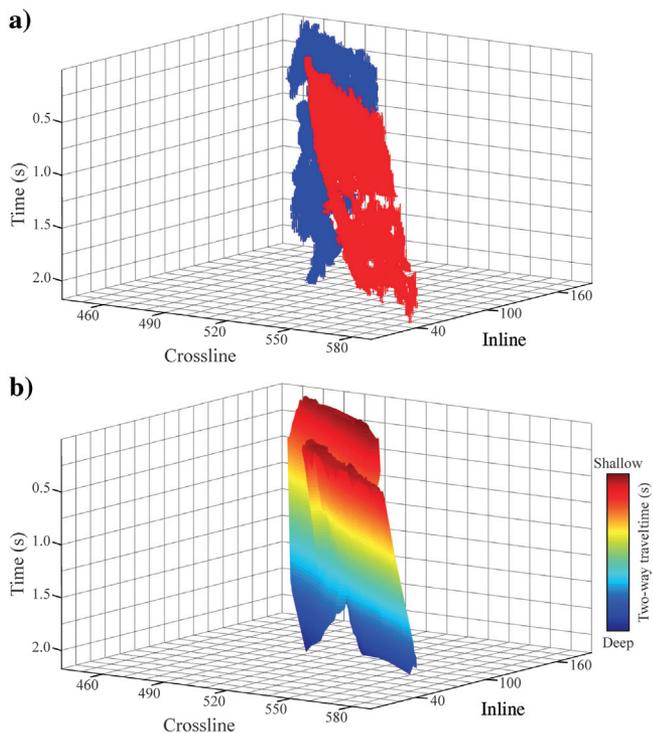


Figure 20. (a) The manually merged fault surface patches and (b) generated fault surfaces.

the conjugate fault system with minimal interpreter intervention. Our workflow may need the manual fault patch merging step only if the fault attribute fails to represent the fault surfaces near the intersection of the fault surfaces.

### Calculating fault parameters using 3D WPCA

We produce fault surfaces using fault sticks, and fault sticks are binarized from fault attributes. Each fault pixel of the fault stick has the same weight on the fault surface construction. However, we propose that the fault pixels that have a higher fault probability should have greater weight in the orientation computation of the fault surfaces. To achieve this goal, we recompute the fault dip and azimuth of each fault pixel using 3D WPCA after we obtain the fault surfaces. We use model coordinate  $(x, y, z)$  to describe the location of each fault pixel located at the 3D fault surface. To compute the fault dip and azimuth of the analyzed fault pixel, we first extract the fault pixels centered at the analyzed pixel. The window used to extract the fault pixels is a cube window, and the size of the window is  $l_{min}$  by  $l_{min}$  by  $l_{min}$  pixels. Supposing that there are  $N$  fault pixels within the analysis window, the coordinate vector of the fault pixels can be expressed as  $\mathbf{x} = (x_1, x_2, \dots, x_N)$ ,  $\mathbf{y} = (y_1, y_2, \dots, y_N)$ , and  $\mathbf{z} = (z_1, z_2, \dots, z_N)$ . The seismic fault attribute values of the fault pixels are expressed as  $\mathbf{s} = (s_1, s_2, \dots, s_N)$ . Then, the covariance matrix  $\mathbf{C}_w$  is defined as

$$\mathbf{C}_w = \frac{1}{N-1} \begin{bmatrix} \text{cov}_w(\mathbf{x}, \mathbf{x}) & \text{cov}_w(\mathbf{x}, \mathbf{y}) & \text{cov}_w(\mathbf{x}, \mathbf{z}) \\ \text{cov}_w(\mathbf{y}, \mathbf{x}) & \text{cov}_w(\mathbf{y}, \mathbf{y}) & \text{cov}_w(\mathbf{y}, \mathbf{z}) \\ \text{cov}_w(\mathbf{z}, \mathbf{x}) & \text{cov}_w(\mathbf{z}, \mathbf{y}) & \text{cov}_w(\mathbf{z}, \mathbf{z}) \end{bmatrix}, \quad (5)$$

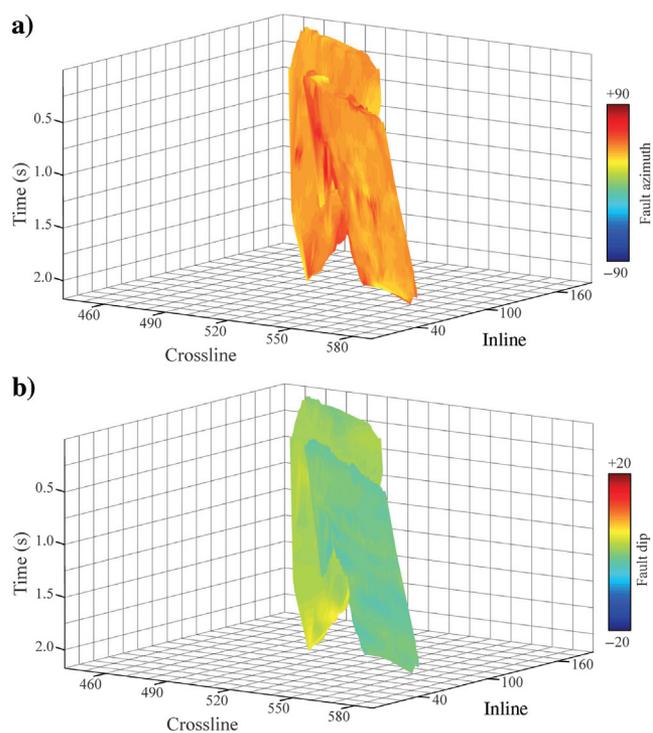


Figure 21. (a) The calculated fault azimuth and (b) fault dip overlaid with the generated fault surfaces in Figure 19b.

where  $\text{cov}_w(\mathbf{x}, \mathbf{x})$ ,  $\text{cov}_w(\mathbf{x}, \mathbf{y})$ ,  $\text{cov}_w(\mathbf{y}, \mathbf{x})$ , and  $\text{cov}_w(\mathbf{y}, \mathbf{y})$  are defined in equation 2 and  $\text{cov}_w(\mathbf{x}, \mathbf{z})$ ,  $\text{cov}_w(\mathbf{y}, \mathbf{z})$ ,  $\text{cov}_w(\mathbf{z}, \mathbf{z})$ ,  $\text{cov}_w(\mathbf{z}, \mathbf{x})$ , and  $\text{cov}_w(\mathbf{z}, \mathbf{y})$  are defined as follows:

$$\text{cov}_w(\mathbf{x}, \mathbf{z}) = \frac{\sum_{i=1}^N s_i (x_i - \bar{x})(z_i - \bar{z})}{N - 1}, \quad (6a)$$

$$\text{cov}_w(\mathbf{y}, \mathbf{z}) = \frac{\sum_{i=1}^N s_i (y_i - \bar{y})(z_i - \bar{z})}{N - 1}, \quad (6b)$$

$$\text{cov}_w(\mathbf{z}, \mathbf{z}) = \frac{\sum_{i=1}^N s_i (z_i - \bar{z})(z_i - \bar{z})}{N - 1}, \quad (6c)$$

$$\text{cov}_w(\mathbf{z}, \mathbf{x}) = \frac{\sum_{i=1}^N s_i (z_i - \bar{z})(x_i - \bar{x})}{N - 1}, \quad (6d)$$

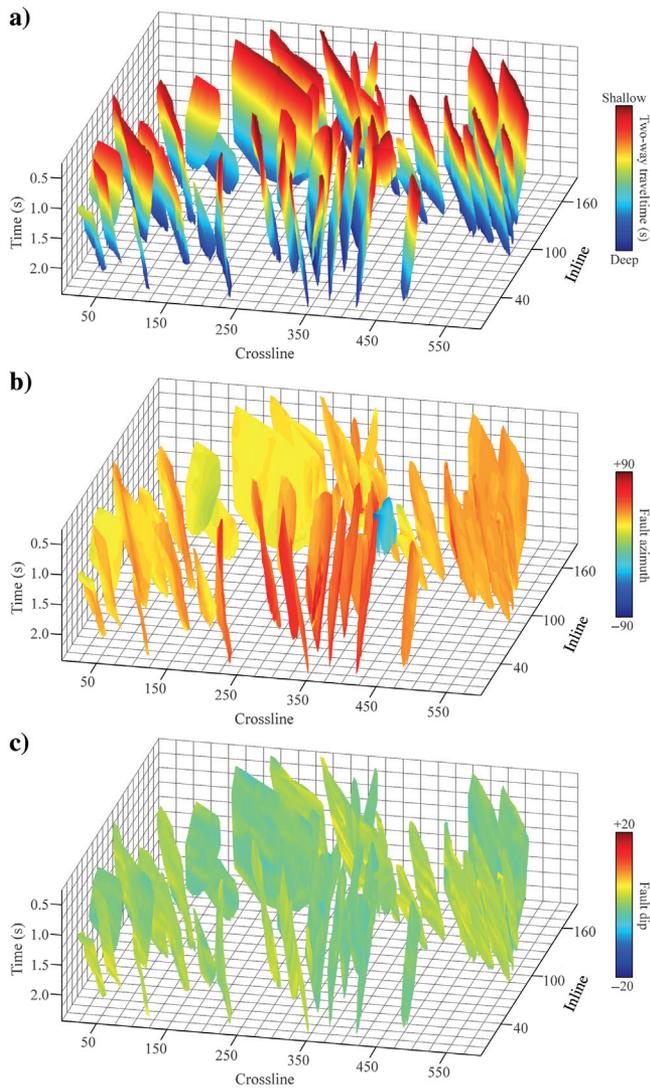


Figure 22. (a) Generated fault surfaces and the corresponding (b) fault azimuth and (c) fault dip.

$$\text{cov}_w(\mathbf{z}, \mathbf{y}) = \frac{\sum_{i=1}^N s_i (z_i - \bar{z})(y_i - \bar{y})}{N - 1}. \quad (6e)$$

We obtain eigenvalues and eigenvectors of the covariance matrix in equation 5 by applying the eigen decomposition

$$\begin{aligned} \mathbf{C}_w &= \lambda_u \mathbf{u}\mathbf{u}^T + \lambda_v \mathbf{v}\mathbf{v}^T + \lambda_w \mathbf{w}\mathbf{w}^T \\ &= \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} \end{bmatrix} \begin{bmatrix} \lambda_u & 0 & 0 \\ 0 & \lambda_v & 0 \\ 0 & 0 & \lambda_w \end{bmatrix} \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}, \end{aligned} \quad (7)$$

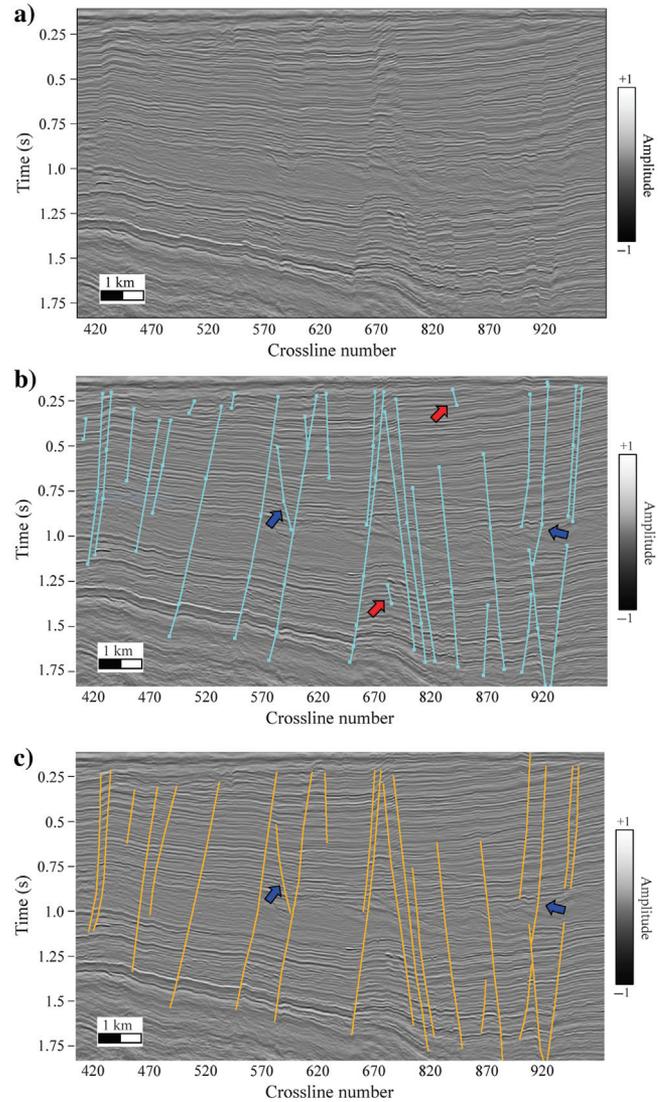


Figure 23. The comparison of automatic and manually interpreted fault sticks on one inline seismic section. (a) One inline seismic section. (b) The manually interpreted fault sticks (the cyan curve) overlaid on the seismic section. The squares are the manually picked control points used to generate the fault sticks. (c) The automatic interpreted fault sticks (the yellow curve) overlaid on the seismic section.

where  $\lambda_u, \lambda_v$ , and  $\lambda_w$  are the eigenvalues satisfied by  $\lambda_u \geq \lambda_v \geq \lambda_w \geq 0$  and  $\mathbf{u}$ ,  $\mathbf{v}$ , and  $\mathbf{w}$  are the corresponding normalized eigenvectors. According to the PCA theory, the eigenvector  $\mathbf{w}$  is perpendicular to the fault surface at the analysis pixel. The fault dip  $d$  and azimuth  $a$  of the analysis pixel are defined as, respectively,

$$d = \arctan\left(\frac{w_z}{\sqrt{w_x^2 + w_y^2}}\right), \quad (8)$$

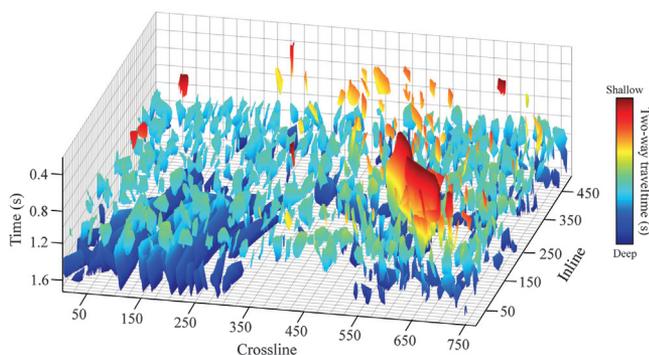


Figure 24. Generated fault surfaces from the Netherlands F3 survey.

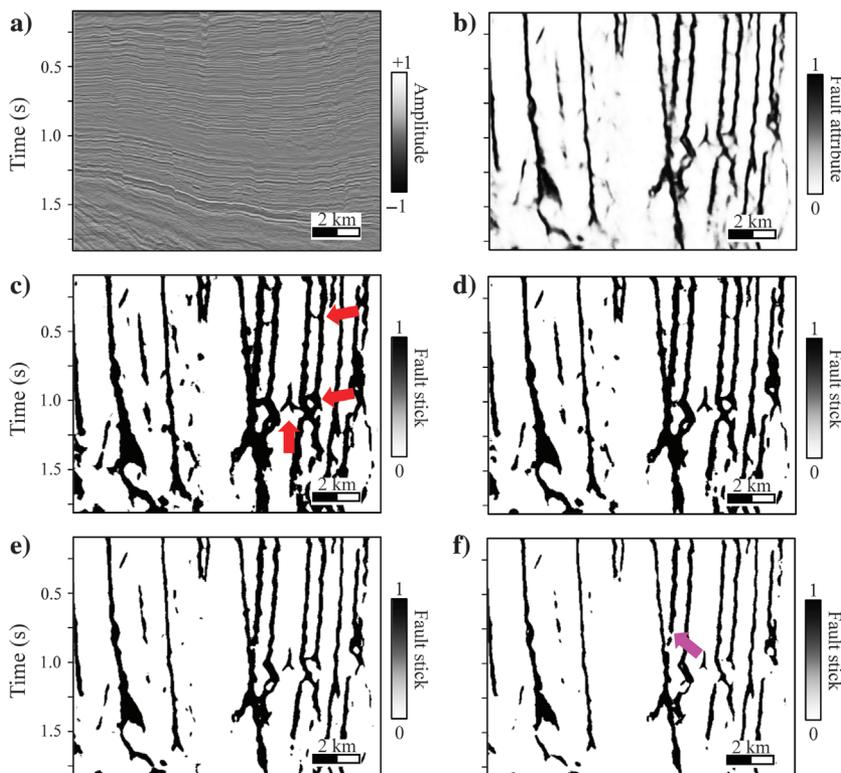


Figure 25. Testing the sensitivity of parameter  $f_{\min}$  on the Kerry seismic survey. (a) One inline seismic section and (b) the corresponding seismic fault attribute. The binarized results using values of (c) 0.1, (d) 0.2, (e) 0.4, and (f) 0.6, respectively. The red arrow indicates the representative locations where possible “wrong” fault pixels were generated by the low value. The purple arrow indicates the representative fault sticks that break apart by a high value.

$$a = \arctan\left(\frac{w_y}{w_x}\right), \quad (9)$$

where  $w_x, w_y$ , and  $w_z$  are the elements of eigenvector  $\mathbf{w}$  in the  $x$ -,  $y$ -, and  $z$ -directions, respectively. The units of  $d$  and  $a$  are degrees.

The calculated fault dip and azimuth vectors for the pixels on the fault surface are denoted as  $\mathbf{d} = (d_1, d_2, \dots, d_N)$  and  $\mathbf{a} = (a_1, a_2, \dots, a_N)$ , respectively. Figure 21a and 21b shows the calculated fault azimuth and dip overlaid on the fault surfaces shown in Figure 20. Note that we have a uniform fault dip and azimuth on the same fault surface.

### 3D fault surfaces

Figure 22a shows the generated representative fault surfaces within the Kerry seismic survey. The parameters  $f_{\min}$ ,  $l_{\min}$ ,  $\theta_{\text{thrd}}$ , and  $s_{\min}$  are set as 0.2, 15,  $5^\circ$ , and 5%, respectively. The value of  $l_{\min}$  is 15, which means that our algorithm will not construct surfaces for those faults whose largest thinned fault stick on time or vertical slices is less than 15 pixels. We generate 192 fault surfaces in this seismic survey, and Figure 22a only shows the biggest 60 fault surfaces. Figure 22b and 22c shows the fault azimuth and dip overlaid with fault surfaces in Figure 22a, respectively. Note that we have uniform fault dip and azimuth on the same fault surface. Figure 23a–23c shows one inline seismic section, manually picked fault sticks overlaid on the inline seismic section, and the automatically picked fault sticks overlaid on the inline seismic section, respectively.

The automatically picked fault sticks overlaid on the inline seismic section, respectively. The automatically picked fault sticks shown in Figure 23c were extracted from the automatically picked fault surfaces. The manually picked fault sticks usually have three to six control points (the squares in Figure 23a). The fault sticks shown in Figure 23b were generated by the interpreting software using a linear interpolation between two nearby control points. Note that the automatically extracted fault sticks strictly follow the fault trace on the inline seismic section. However, our algorithm fails to extract the small fault sticks (indicated by the red arrows) whose length is smaller than the threshold  $l_{\min}$  (the smallest length of the fault sticks on vertical section). Note that our algorithm produces better fault sticks (indicated by the blue arrows) than shown in Figure 23b. Of course, we can always improve the results shown in Figure 23b by adding more control points.

To further demonstrate the reproducibility of the proposed method, we apply our proposed workflow to another 3D marine seismic survey, the F3. The coherence threshold  $f_{\min}$ , the smallest length  $l_{\min}$ , the orientation threshold  $\theta_{\text{thrd}}$ , and the percentage of exclusive fault sticks  $s_{\min}$  are set as 0.2, 15,  $5^\circ$ , and 5%, respectively. The parameters used in the F3 seismic survey are the same as the parameters set in the Kerry seismic survey. Figure 24 shows the generated representative fault surfaces within the F3 seismic survey. We generate 422 fault surfaces in this

seismic survey, and Figure 24 only shows the largest 250 fault surfaces.

## DISCUSSION

The proposed workflow needs interpreters defining four parameters: (1) the fault attribute threshold  $f_{\min}$  used to generate fault sticks, (2) the smallest length  $l_{\min}$  in pixels of the thinned fault sticks we want to detect on time or vertical slices, (3) the orientation threshold  $\theta_{\text{thrd}}$  used to refine the fault sticks, and (4) the percentage of exclusive fault sticks  $s_{\min}$  used in the surface patch merging step. Figures 25 and 26 illustrate the sensitivity testing of the fault attribute threshold  $f_{\min}$ . The fault attribute value of the Kerry and F3 seismic surveys is normalized to the 0–1 range, where value 1 stands for the highest possibility of being a fault. Figure 25a and 25b shows one inline seismic section of the Kerry seismic survey and the fault attribute section. Figure 25c–25f shows the corresponding binarized results using threshold value  $f_{\min}$  of 0.1, 0.2, 0.4, and 0.6, respectively. Figure 26a and 26b shows one inline seismic section of the F3 seismic survey and the fault attribute section. Figure 26c–26f shows the corresponding binarized results using the threshold value of 0.1, 0.2, 0.4, and 0.6, respectively. Figures 25c and 26c indicate that it is prone to generate “noisy” fault pixels (indicated by the red arrows in Figures 25f and 26f) if  $f_{\min} < 0.2$ . The noisy pixels would increase the computing in the following process of fault stick generating. Figures 25 and 26 indicate that it is prone to break apart the fault pixels (indicated by the purple arrow) that belong to the same fault stick. Thus, we suggest the value of  $f_{\min}$  to be set as a value between  $0.2f_{\max}$  and  $0.4f_{\max}$ , where  $f_{\max}$  is the maximum value of fault probability.

The smallest length  $l_{\min}$  in pixels is determined by the smallest fault surface size that we want to identify within the seismic survey. Our workflow will ignore those fault surfaces whose largest fault stick size on time or vertical slices is smaller than  $l_{\min}$ . Note that the size of one pixel along the vertical axis is the same length of time sample interval. Similarly, the sizes of one pixel along the inline and crossline directions are the same length of seismic trace intervals along the inline and crossline directions, respectively. Figure 27a and 27b shows one inline section of the Kerry seismic survey and the corresponding binarized result. Figure 27c and 27d shows the fault sticks after applying the threshold  $l_{\min}$  of 10 and 20, respectively.

A small value of threshold  $\theta_{\text{thrd}}$  reduces the time of generating fault surfaces but increases the time of merging fault surfaces. Considering that we have an optional step of manual merging, we suggest using a small value of threshold  $\theta_{\text{thrd}}$ ; a small value can guarantee the accuracy of the generated fault surface patches. Figure 28a and

28b shows the fault attribute and the corresponding binarized result. Figure 28c and 28d shows the fault sticks after applying threshold  $\theta_{\text{thrd}}$  of  $5^\circ$  and  $10^\circ$ , respectively. Figure 28e illustrates the strategy we use to define the dip of the fault stick in this paper. The red and blue dots indicate the fault and nonfault pixels of the seismic traces, respectively. Then, we compute the dip of the fault stick by using  $\arctan(\frac{\Delta t}{l})$ , where  $\Delta t$  is the time interval (in the unit of the sample number) between the fault pixels of the seismic traces and  $l$  is the horizontal interval in the unit of the trace number.

Figure 29 shows the sensitivity testing of parameter  $s_{\min}$ . Figure 29a and 29b shows the generated fault surface patches by using threshold  $s_{\min}$  of 1% and 10%, respectively. Compared to the result shown in Figure 18, our algorithm produced more small patches (indicated by the arrows in Figure 29a) if we set  $s_{\min}$  to 1%. Note

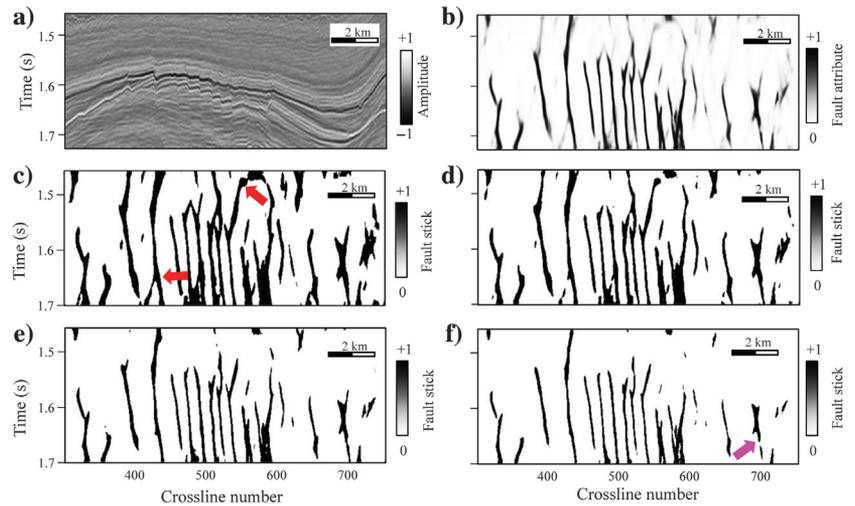


Figure 26. Testing the sensitivity of parameter  $f_{\min}$  on the F3 seismic survey. (a) One inline seismic section and (b) the corresponding seismic fault attribute. The binarized results using values of (c) 0.1, (d) 0.2, (e) 0.4, and (f) 0.6, respectively. The red arrow indicates the representative locations where possible “wrong” fault pixels were generated by the low value. The purple arrow indicates representative fault sticks that break apart by a high value.

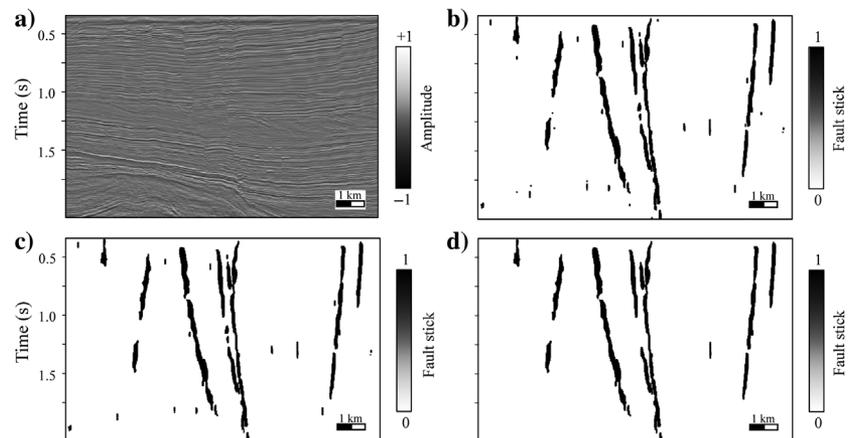


Figure 27. Testing the sensitivity of parameter  $l_{\min}$  on the Kerry seismic survey. (a) One inline seismic section and (b) the corresponding binarized fault attribute. The processed fault sticks using values of (c) 10 and (d) 20, respectively.

that our algorithm generated large corrected fault surface patches (indicated by the blue arrow in Figure 29b) if we set  $s_{\min}$  to 10%. However, we find that it is much easier merging fault surface patches than separating wrongly merged fault surface patches. Thus, we suggest setting  $s_{\min}$  at relatively low value (e.g., smaller than 10%). Of course, we need interpreters testing those parameters on a case-by-case basis.

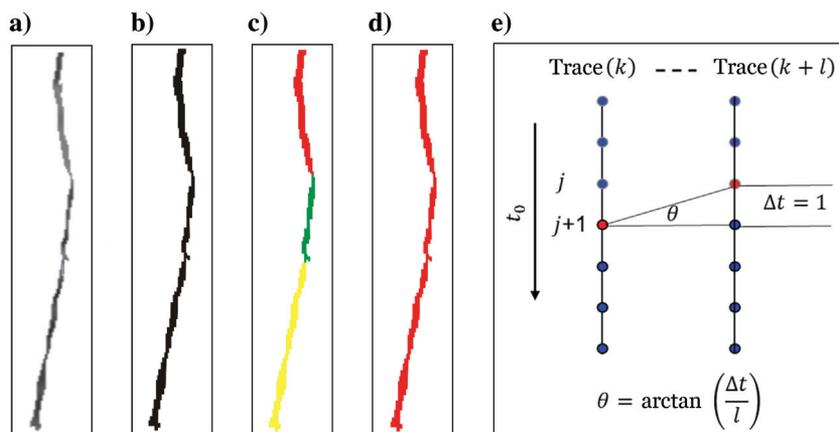


Figure 28. Testing the sensitivity of parameter  $\theta_{\text{thrd}}$  on the Kerry seismic survey. (a) The seismic fault attributes, (b) the binarized result, (c) the processed fault stick using the value of  $5^\circ$ , (d) the processed fault stick using the value of  $10^\circ$ , and (e) illustration of the strategy of computing the dip of the fault stick.

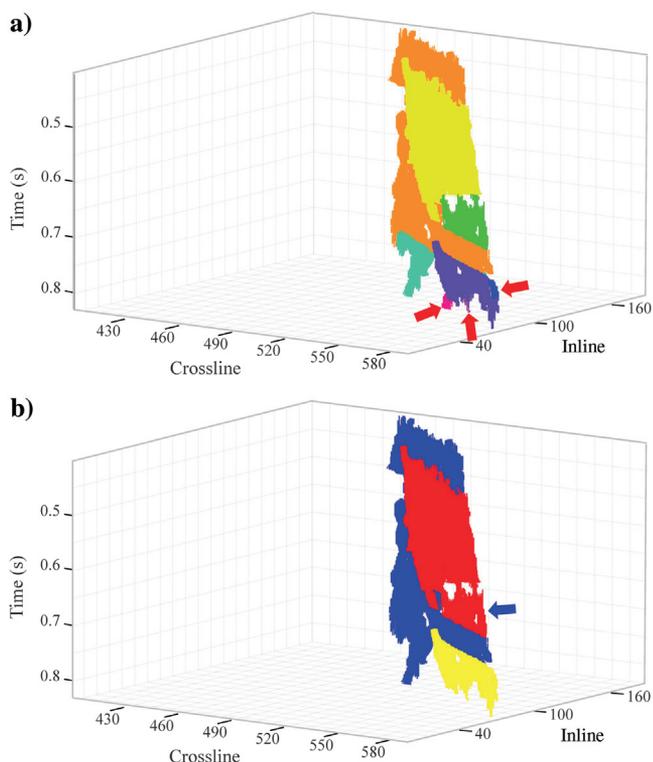


Figure 29. Testing the sensitivity of parameter  $s_{\min}$ : (a) The produced fault surface patches using the value of 1%. The red arrows indicate the small fault surface patches. (b) The produced fault surface patches using the value of 10%. The blue arrow indicates a large fault surface patch.

An experienced interpreter needs approximately 30 min to pick the fault sticks shown in Figure 23b. It may take weeks to manually pick all of the fault sticks considering that the Kerry seismic survey contains hundreds of inline slices. The Kerry seismic data were processed on a computer with two nodes (Intel Xeon CPU E5-2670, 2.3 GHz). Each node contains 24 processors. The entire code was run using the MATLAB Parallel Computing Toolbox. The

computation times of generating fault sticks and grouping are approximately 4 and 6 h, respectively. It takes 4 h of interpreter work (merging the fault surface patches and quality control) for producing all of the fault surfaces (192) by using our method. Note that our algorithm does not extract the fault surface if the fault stick on the vertical slice is smaller than approximately 60 ms. Thus, we conclude that our method accelerates the procedure of seismic fault interpretation.

## CONCLUSION

We proposed a novel workflow to semiautomatically generate fault surfaces from seismic fault attributes. Our workflow is based on analyzing the topological relationship (connectivity and mutual exclusion) in a 3D manner among the fault sticks on time and vertical slices. The quality of the input seismic fault attribute is one of the

key factors for successfully implementing our workflow. Thus, we recommend conditioning the seismic fault attribute prior to applying our workflow if the seismic fault attribute has strong staircase artifacts and undesired stratigraphic anomalies. The deep-learning-based fault attribute algorithms have produced high-quality seismic fault attributes without staircase artifacts and undesired stratigraphic anomalies. Thus, we suggest using fault attributes generated using the deep-learning algorithm as the input for our algorithm. The computation cost of our algorithm depends on the complexity of the fault system and the noise level of the seismic data. Complex faults such as conjugate faults or high-level noise would result in more fault surface patches and would need more human intervention.

## DATA AND MATERIALS AVAILABILITY

Data associated with this research are available and can be accessed via the following URL: [https://wiki.seg.org/wiki/Open\\_data](https://wiki.seg.org/wiki/Open_data).

## REFERENCES

- AlBinHassan, N., and K. Marfurt, 2003, Fault detection using Hough transforms: 73rd Annual International Meeting, SEG, Expanded Abstracts, 1719–1721, doi: [10.1190/1.1817639](https://doi.org/10.1190/1.1817639).
- Barnes, A. E., 2007, A tutorial on complex seismic trace analysis: *Geophysics*, **72**, no. 6, W33–W43, doi: [10.1190/1.2785048](https://doi.org/10.1190/1.2785048).
- Cohen, I., N. Coult, and A. A. Vassiliou, 2006, Detection and extraction of fault surfaces in 3D seismic data: *Geophysics*, **71**, no. 4, P21–P27, doi: [10.1190/1.2215357](https://doi.org/10.1190/1.2215357).
- dGB Earth Sciences, 1987, Netherlands offshore F3 block complete, <https://terranubis.com/datainfo/Netherlands-Offshore-F3-Block-Complete>, accessed 20 January 2019.
- Fan, Z., E. Liu, and B. Xu, 2011, Weighted principal component analysis: International Conference on Artificial Intelligence and Computational Intelligence, 569–574.

- 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](https://doi.org/10.1190/1.1444651).
- Gonzalez, R., and R. Woods, 1992, *Digital image processing*: Addison-Wesley Publishing Company.
- Hale, D., 2009, Structure-oriented smoothing and semblance: CWP Report, 635, 261–270.
- Lou, Y., B. Zhang, R. Wang, T. Lin, and D. Cao, 2019, Seismic fault attribute estimation using a local fault model: *Geophysics*, **84**, no. 4, O73–O80, doi: [10.1190/geo2018-0678.1](https://doi.org/10.1190/geo2018-0678.1).
- Marfurt, K. J., 2006, Robust estimates of 3D reflector dip and azimuth: *Geophysics*, **71**, no. 4, P29–P40, doi: [10.1190/1.2213049](https://doi.org/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](https://doi.org/10.1190/1.1444415).
- Neff, D. B., J. R. Grismore, and W. A. Lucas, 2000, Automated seismic fault detection and picking: U.S. Patent 6,018,498.
- Pedersen, S. I., T. Randen, L. Sønneland, and Ø. Steen, 2002, Automatic fault extraction using artificial ants: 72nd Annual International Meeting, SEG, Expanded Abstracts, 512–515, doi: [10.1190/1.1817297](https://doi.org/10.1190/1.1817297).
- Pedersen, S. I., T. Skov, A. Hettelid, P. Fayemendy, T. Randen, and L. Sønneland, 2003, New paradigm of fault interpretation: 73rd Annual International Meeting, SEG, Expanded Abstracts, 350–353, doi: [10.1190/1.1817918](https://doi.org/10.1190/1.1817918).
- Qi, J., and J. Castagna, 2013, Application of a PCA fault-attribute and spectral decomposition in Barnett Shale fault detection: 83rd Annual International Meeting, SEG, Expanded Abstracts, 1421–1425, doi: [10.1190/segam2013-0674.1](https://doi.org/10.1190/segam2013-0674.1).
- Qi, J., B. Lyu, A. AlAli, G. Machado, Y. Hu, and K. J. Marfurt, 2018, Image processing of seismic attributes for automatic fault extraction: *Geophysics*, **84**, no. 1, O25–O37, doi: [10.1190/geo2018-0369.1](https://doi.org/10.1190/geo2018-0369.1).
- Qi, J., G. Machado, and K. J. Marfurt, 2017, A workflow to skeletonize faults and stratigraphic features: *Geophysics*, **82**, no. 4, O57–O70, doi: [10.1190/geo2016-0641.1](https://doi.org/10.1190/geo2016-0641.1).
- Wu, X., and S. Fomel, 2018, Automatic fault interpretation with optimal surface voting: *Geophysics*, **83**, no. 5, O67–O82, doi: [10.1190/geo2018-0115.1](https://doi.org/10.1190/geo2018-0115.1).
- Wu, X., and D. Hale, 2016, 3D seismic image processing for faults: *Geophysics*, **81**, no. 2, IM1–IM11, doi: [10.1190/geo2015-0380.1](https://doi.org/10.1190/geo2015-0380.1).
- Zhang, B., Y. Liu, M. Pelissier, and N. Hemstra, 2014, Semiautomated fault interpretation based on seismic attributes: *Interpretation*, **2**, no. 1, SA11–SA19, doi: [10.1190/INT-2013-0060.1](https://doi.org/10.1190/INT-2013-0060.1).

Biographies and photographs of the authors are not available.