GEOPHYSICS, VOL. 81, NO. 2 (MARCH-APRIL 2016); P. IM1–IM11, 11 FIGS.
10.1190/GEO2015-0380.1

3D seismic image processing for faults

Xinming Wu[1] and Dave Hale[1]

ABSTRACT

Numerous methods have been proposed to automatically extract fault surfaces from 3D seismic images, and those surfaces are often represented by meshes of triangles or quadrilaterals. However, extraction of intersecting faults is still a difficult problem that is not well addressed. Moreover, mesh data structures are more complex than the arrays used to represent seismic images, and they are more complex than necessary for subsequent processing tasks, such as that of automatically estimating fault slip vectors. We have represented a fault surface using a simpler linked data structure, in which each sample of a fault corresponded to exactly one seismic image sample, and the fault samples were linked above and below in the fault dip directions, and left and right in the fault strike directions. This linked data structure was easy to exchange between computers and facilitated subsequent image processing for faults. We then developed a method to construct complete fault surfaces without holes using this simple data structure and to extract multiple intersecting fault surfaces from 3D seismic images. Finally, we used the same structure in subsequent processing to estimate fault slip vectors and to assess the accuracy of estimated slips by unfaulting the seismic images.

INTRODUCTION

Faults such as those shown in Figure 1 are important geologic surfaces that we can automatically extract from 3D seismic images. When extracting a fault surface, we also want to obtain fault strikes, dips, and slip vectors, as illustrated in Figure 2.

To extract fault surfaces, fault images (like that shown in Figure 1a) are first computed from a seismic image. These fault images indicate where faults might exist. Many methods have been developed to compute fault images using attributes such as semblance (Marfurt et al., 1998), coherency (Marfurt et al., 1999), variance (Van Bemmel and Pepper, 2000; Randen et al., 2001), gradient magnitude (Aqrawi and Boe, 2011), and fault likelihood (Hale, 2013b).

Various methods have also been proposed to extract fault surfaces from computed fault images, Pedersen et al. (2002) and Pedersen et al. (2003) propose the ant-tracking method to first extract small fault segments, which are then merged to form larger fault surfaces. Similarly, the methods proposed by Gibson et al. (2005), Admasu et al. (2006), and Kadlec et al. (2008) also try to build larger fault surfaces from smaller patches. Hale (2013b) uses images of fault likelihoods, strikes, and dips to construct fault surfaces that coincide with ridges of the fault likelihood image.

From extracted fault surfaces, fault slips can be estimated by correlating seismic reflectors or picked horizons on opposite sides of the fault surfaces. For example, Borgos et al. (2003) correlate seismic horizons across faults by using a clustering method with multiple seismic attributes. Admasu (2008) uses Bayesian matching of seismic horizons extracted on opposite sides of faults. Aurnhammer and Tonnies (2005) and Liang et al. (2010) use windowed cross-correlation methods to correlate seismic reflectors across faults. Hale (2013b) uses a dynamic image warping method.

As discussed above, various methods have been proposed to compute fault images, extract fault surfaces, and estimate fault slips. However, the problem of extracting intersecting faults, such as those shown in Figure 1, is not well addressed. For example, the method described by Hale (2013b) assumes that a single seismic image sample can be associated with only one fault, and it therefore extracts incomplete fault surfaces, with holes at the intersections. Incomplete fault surfaces cause problems for all the above methods used to estimate fault slips because near holes it is difficult to determine which seismic reflectors should be correlated.

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

Manuscript received by the Editor 13 July 2015; revised manuscript received 3 September 2015; published online 26 February 2016; corrected version published online 15 March 2016.
[1]Colorado School of Mines, Golden, Colorado, USA. E-mail: xinwu@mines.edu; iradavidhale@gmail.com.

We first use the method described by Hale (2013b) to compute images of fault likelihoods, strikes, and dips. Each of these images has nonzero values only at faults, as in the fault likelihood image shown in Figure 1a. Therefore, these three fault images can be represented, all at once, by the fault samples shown in Figure 1b. Each fault sample corresponds to one and only one seismic image sample, and it can be displayed as a small square colored by the fault likelihood and oriented by strike and dip.

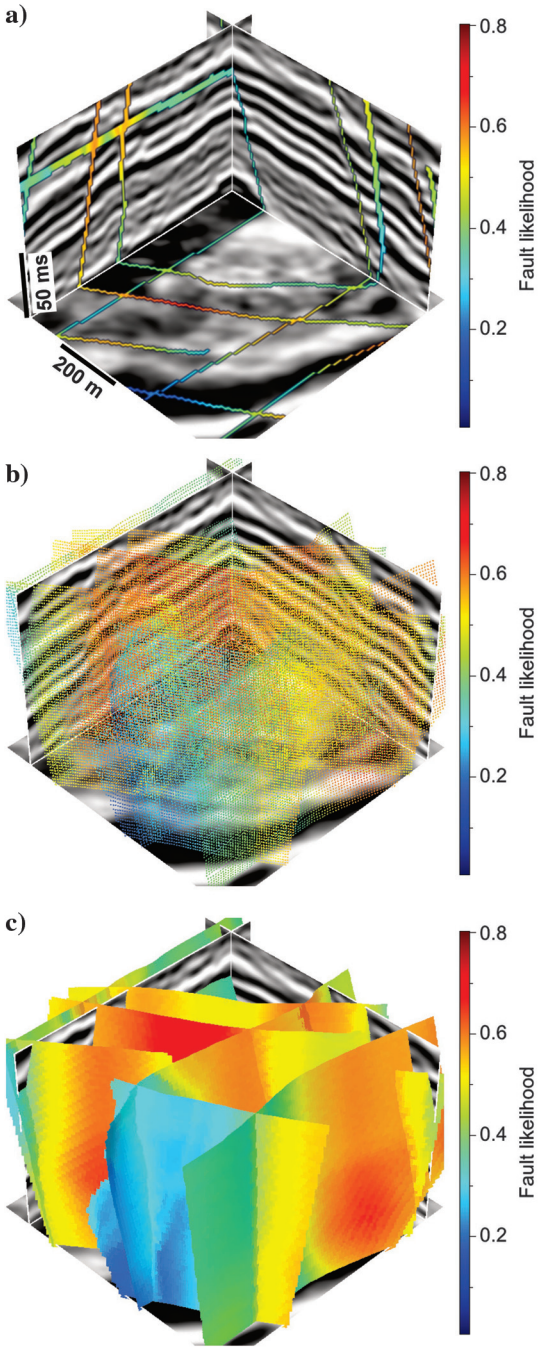

Figure 1. A small subset of 3D seismic image displayed with (a) a fault image, (b) fault samples, (c) fault surfaces, and all faults are colored by fault likelihood.

We then use the fault samples in Figure 1b to construct fault surfaces, which appear to be continuous, as shown in Figure 1c, but they are actually only linked sets of the fault samples in Figure 1b. In Figure 1c, we simply increase the size of squares that are used to represent fault samples, so that they overlap and appear to form continuous surfaces. Each of these fault surfaces is constructed by linking each fault sample with neighbors above, below, left, and right. If any of the four neighbors are missing, we attempt to create them using a method proposed in this paper. In this way, we fill holes and merge separated fault segments to form more complete and intersecting faults as shown in Figure 1c. With more complete surfaces without holes, we are able to more accurately estimate fault slips. To verify the estimated slips, we use them in an unfaulting processing to correlate seismic reflectors across faults.

FAULT IMAGES

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

In 3D seismic images such as those shown in Figures 1 and 3, faults appear as discontinuities that are locally planar (or locally linear in 2D slices). This means that to highlight faults from a seismic image, we do not only look for discontinuities, but rather for discontinuities that are locally planar. Fault likelihood, as defined by Hale (2013b), is one such measure of locally planar discontinuity. Therefore, we use Hale's (2013b) method to compute the fault likelihood (Figure 4a), whereas at the same time estimating the fault strike (Figure 4b) and dip (Figure 4c). The fault likelihood image indicates where faults might exist, whereas the strike and dip images indicate their orientations.

In computing these fault images, this method scans over a range of possible combinations of strike and dip to find the one orientation

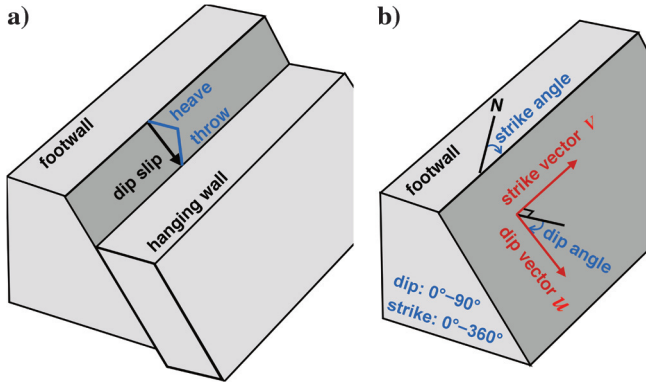

Figure 2. (a) Fault dip slip is a vector representing displacement in the dip direction, of the hanging wall side of a fault surface relative to the footwall side. Fault throw is the vertical component of the slip. Fault strike and dip angles with corresponding unit vectors are defined in panel (b).

that maximizes the fault likelihood for each image sample (Hale, 2013b). The maximum fault likelihood for each image sample is recorded in the fault likelihood image (Figure 4a), and the strike and dip angles that yield the maximum likelihood are recorded in the fault strike (Figure 4b) and dip (Figure 4c) images, respectively. In the fault likelihood image in Figure 4a, we expect relatively high values in areas in which faults might exist. In the fault strike (Figure 4b) and dip (Figure 4c) images, we expect the strike and dip angles to be accurate only where faults are likely, that is, where fault likelihoods are high.

As discussed by Hale (2013b), a significant limitation of this scanning method is in dealing with intersecting faults. Because only a single fault likelihood value, and its corresponding fault strike and dip are recorded for each image sample, this method implicitly assumes that each image sample can be associated with only one fault. This assumption is not valid for samples in which two or more faults intersect. For example, in the intersection area highlighted by cycles in Figure 4, fault likelihoods, strikes, and dips only for fault F-B have been recorded, and the information corresponding to fault F-A is missing. Fault likelihoods of fault F-A might also be high near this intersection, but they have been discarded together with corresponding strikes and dips, only because they were smaller than the fault likelihoods computed for fault F-B. Therefore, fault surfaces directly extracted from such fault images often have holes, especially near fault intersections. We describe below a method to fill holes when constructing fault surfaces.

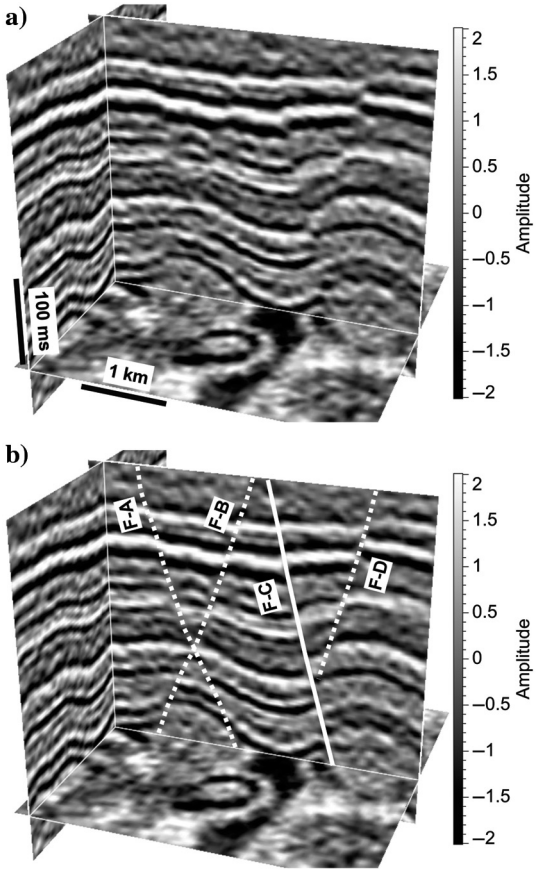

Figure 3. A 3D synthetic seismic image (a) with four faults manually interpreted in panel (b). The dashed lines in panel (b) represent normal faults, whereas the solid line represents a reverse fault.

We do not expect faults to be as thick as the features apparent in the fault likelihood image in Figure 4a. Therefore, we keep only the values on the ridges of the fault likelihood, and we set values elsewhere to be zero to obtain the thinned fault likelihood image shown in Figure 5a. We also keep strike and dip angles for only the samples with nonzero values in Figure 5a to obtain the corresponding thinned fault strike (Figure 5b) and dip (Figure 5c) images. These thinned fault images have nonzero values only for samples that might be on faults.

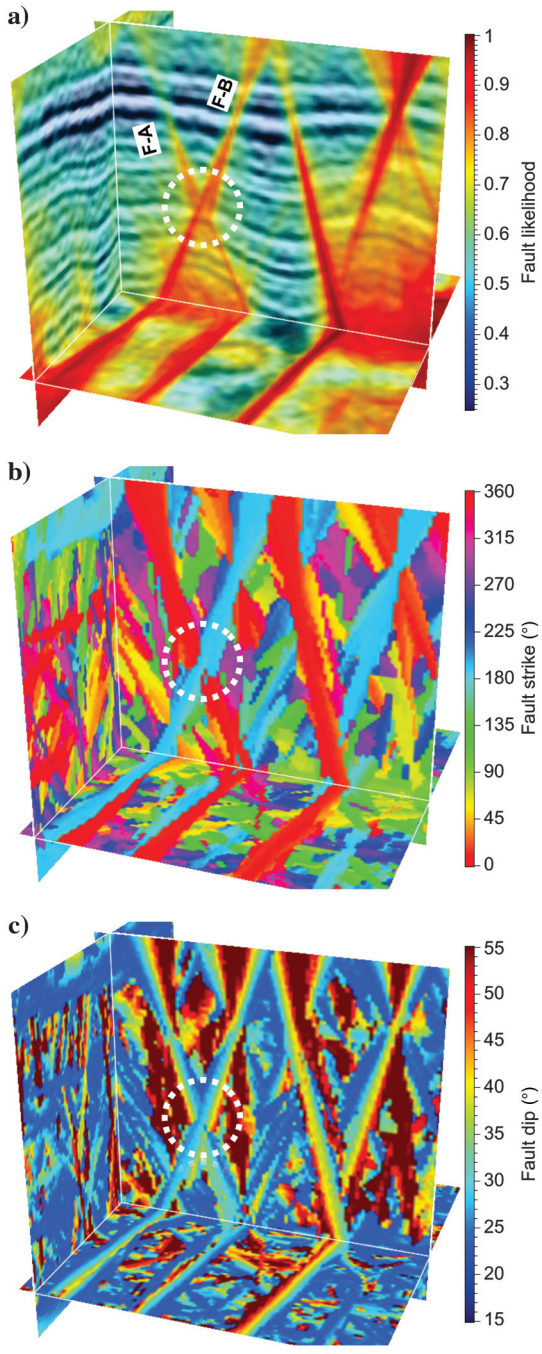

Figure 4. A 3D seismic image with (a) fault likelihoods, (b) strikes, and (c) dips displayed in color. The dashed white circle in each image indicates one location at which fault F-A intersects fault F-B.

We again observe that fault likelihoods, strikes, and dips of fault F-A are missing in the intersection area (the dashed white circles in Figure 5) because of the limitation discussed above. We also observe that some nonzero values appear where faults do not exist. The reason for this is that, in the scanning process, we assume only that faults are locally planar discontinuities. We will discuss below additional conditions that must be satisfied for the nonzero samples in Figure 5 to be considered as faults.

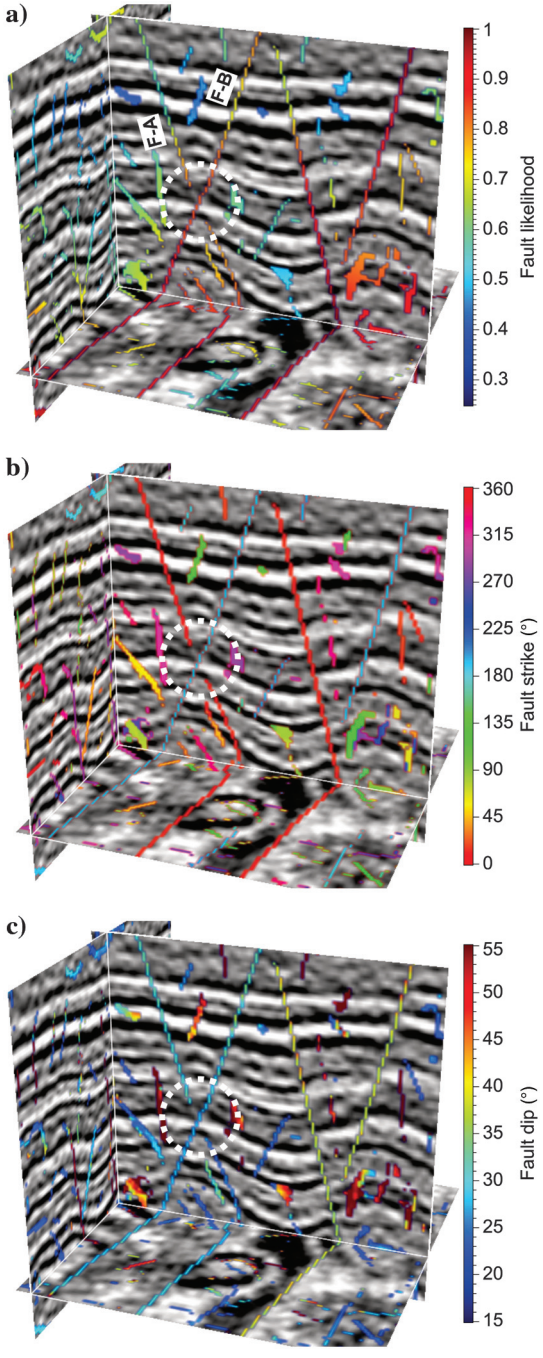

Figure 5. (a) The thinned fault likelihood image has nonzero values only on the ridges of the fault likelihood image in Figure 4a. Fault strikes and dips corresponding to the fault likelihoods are displayed in panels (b and c), respectively. The dashed white circle in each image indicates one location where fault F-A intersects fault F-B.

FAULT SAMPLES AND SURFACES

Notice that most samples in the thinned fault images shown in Figure 5 are zero. For this reason, we use fault samples to more efficiently represent the three fault images with less computer memory. We then extract fault surfaces from the fault samples, and we represent the surfaces using simple and convenient data structures.

Fault samples

Because most samples in the images of the fault likelihood (Figure 5a), strike (Figure 5b), and dip (Figure 5c) are zero, we can display the three images, all at once, as fault samples shown in Figure 6a and more clearly in Figure 7a. Each fault sample is displayed as a colored square. The color of each square denotes the fault likelihood, whereas the orientation of each square represents the fault strike and dip. Fault samples exist only at positions at which thinned fault likelihoods are nonzero, and each fault sample corresponds to one and only one image sample. Therefore, these fault samples contain exactly the same information represented in the thinned fault images shown in Figure 5.

Most fault samples, especially those with high fault likelihoods, are aligned approximate planes, consistent with locally planar fault surfaces. Some misaligned fault samples, often with low fault likelihoods, are also observed in Figures 6a and 7a. These misaligned samples, however, cannot be linked together to form locally planar fault surfaces of significant size.

Fault surfaces

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

For example, to estimate fault slips, we must analyze seismic image samples alongside faults. This means that we must know how to walk vertically up and down (tangent to fault dip), and horizontally left and right (tangent to fault strike) on a fault surface, and thereby to access seismic image samples adjacent to the fault. The quadrilateral mesh discussed by Hale (2013b) is one way to efficiently access seismic image samples alongside a fault. In this paper, we use a simpler linked data structure, shown in Figures 6b, 7b, and 7c, to find seismic image samples adjacent to faults.

Linking fault sample neighbors

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

To find a neighbor above, we need only to consider the upper nine adjacent points in the $3 \times 3 \times 3$ cube of grid points. Among

these nine grid points, we search for a fault sample that lies nearest to the line defined by the center fault sample and its dip vector. Similarly, we search for a neighbor below among the lower nine adjacent grid points.

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

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

The processing above is repeated for each fault sample neighbor until no more neighbors can be found to obtain a set of linked fault samples (Figure 7b). Then, a new seed with a sufficiently high fault likelihood is chosen from unlinked samples for growing a new fault surface. This process ends when no remaining unlinked fault samples have a sufficiently high fault likelihood.

Some samples linked in this way may not correspond to faults. We discard surfaces with small numbers of linked samples, and we keep only those with significant sizes. For example, in Figure 7b, we have kept only the three largest surfaces constructed from the fault samples in Figure 7a. Other fault samples (such as these colored in green and blue in Figure 7a) are then ignored in subsequent processing. The selection of faults based on their size is just one simple way of filtering out spurious faults after constructing fault surfaces. Some geologically reasonable but small faults might be removed by this filtering. We could also construct more complicated filters for fault selection based on the computed fault likelihoods, strikes, dips, and slips.

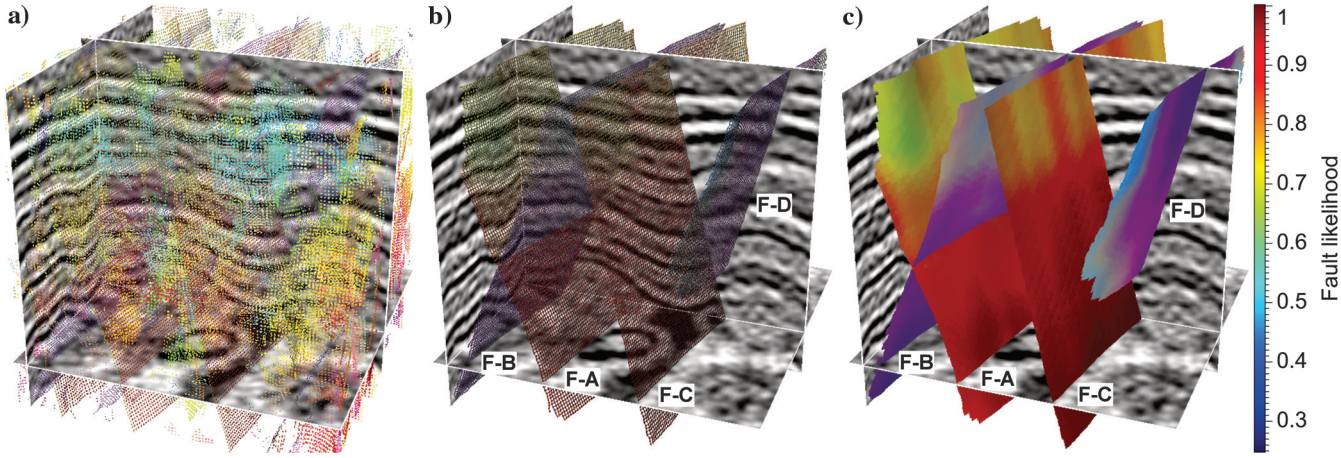

Figure 6. (a) The three fault images in Figure 5 can be represented by fault samples displayed as small squares. Each square in panel (a) is colored by the fault likelihood and oriented by the strike and dip of the corresponding fault sample. Links are then built among consistent fault samples, and each set of linked fault samples in panel (b) represents a fault surface that appears opaque in panel (c), in which fault samples are displayed as larger overlapping squares.

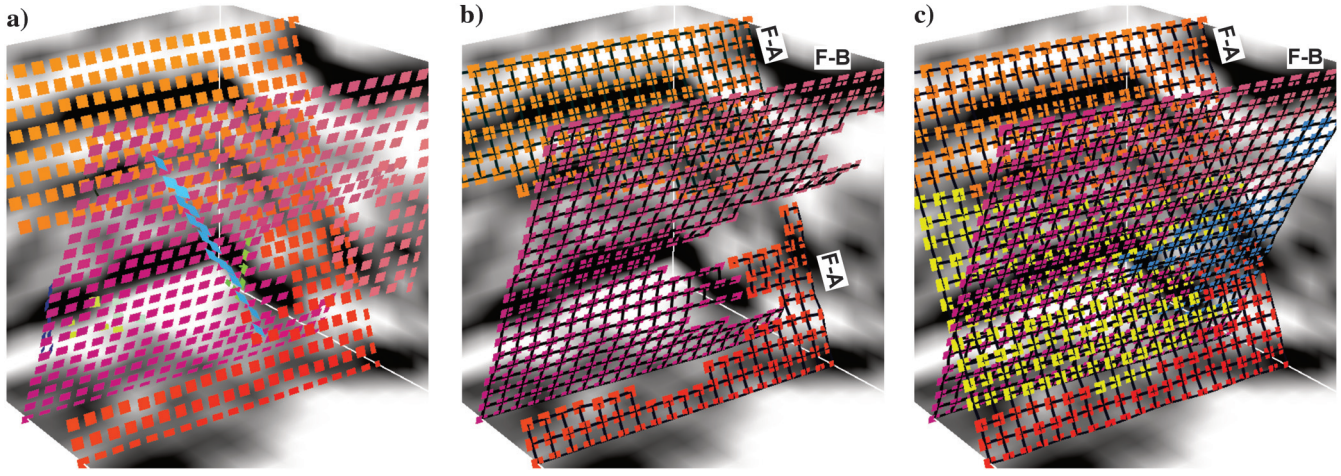

Figure 7. (a) Close-up view of a subset of fault samples from Figure 6a. (b) Links built among nearby fault samples form three sets of linked samples, which represent three fault surfaces (or patches). Near the intersection of faults F-A and F-B, fault F-A is separated into two independent patches, and fault F-B has a hole. (c) New fault samples (colored with yellow and blue) are created to merge the fault patches and fill the hole to construct more complete intersecting fault surfaces.

As shown in Figure 7b, each sample in a fault surface is linked to up to four neighbors. Some neighbors might be missing, and this is in fact necessary because faults are not perfectly aligned with the sampling grid of a 3D seismic image and also because faults are not strictly planar surfaces.

However, some neighbors may be missing because of noise and fault intersections apparent in the seismic image. Where faults intersect, fault samples constructed directly from fault images may be missing, as shown in Figure 7a. These missing fault samples can cause holes within a fault surface, such as fault F-B shown in Figure 7b, and can yield gaps that separate a fault surface into independent patches, like those of fault F-A shown in Figure 7b. To fill in these holes and gaps to construct more complete fault surfaces, we must interpolate missing fault samples, as shown in Figure 7c.

Interpolating missing neighbors

During the processing discussed above for linking neighbors to a fault sample, if any of the neighbors above, below, left, or right are missing, we try to create them. We do not first construct fault surfaces or patches with holes (missing neighbors) as shown in Figure 7b and then fill holes in each of the constructed fault surfaces or patches because in this way we cannot merge fault patches to form a more complete fault surface. Instead, we check for missing neighbors, and we create them as we grow fault surfaces, and we thereby directly obtain complete fault surfaces without holes as shown in Figure 7c.

Remember that each fault sample contains three attributes: fault likelihood, strike, and dip. This means that if we want to create a missing fault sample, we must know not only its position but also its corresponding fault attributes. Therefore, instead of directly creating fault samples, we first construct three fault images of fault likelihood, strike, and dip. Then, we create fault samples from these images. Recall that we find neighbors for a fault sample from only the adjacent grid points, which are in a $3\times3\times3$ cube that is centered at that fault sample. This means that to create a missing neighbor, we need only to create adjacent fault samples and then determine whether any of them could be the missing neighbor.

To create fault samples within a $3\times3\times3$ cube, we must create fault images in a slightly larger cube because fault samples are located on the ridges in a fault likelihood image and additional image samples are needed to find these ridges. Therefore, we construct the new small images of fault likelihood, strike, and dip in a $5\times5\times5$ cube.

To construct a fault likelihood image in a $5\times5\times5$ cube centered at the fault sample with missing neighbors, we first search nearby to find fault samples that have fault attributes similar to those for the center sample. For all examples in this paper, we search for these nearby samples in a $31\times31\times31$ cube. The size of this search window depends on the half-width of the anisotropic Gaussian used below for interpolating fault samples. Suppose that we find N existing fault samples, we then construct a $5\times5\times5$ fault likelihood image by accumulating weighted anisotropic Gaussian functions generated from the N existing fault samples found nearby:

$$f(\mathbf{x}_i)=\sum_{k=1}^{N} f(\mathbf{x}_k)g(\mathbf{x}_k-\mathbf{x}_i), \tag{1}$$

where $f(\mathbf{x}_i)$ denotes a fault likelihood value computed for the ith grid point in the $5\times5\times5$ cube and $\mathbf{x}_i$ denotes the position of that grid point. Here, $f(\mathbf{x}_k)$ denotes the known fault likelihood of the kth nearby fault sample and $g(\mathbf{x})$ is an anisotropic Gaussian function computed for this kth fault sample, given by the following equation:

$$g(\mathbf{x})=\exp\left(-\frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{R}^{\mathsf{T}}\mathbf{S}\mathbf{R}\mathbf{x}\right). \tag{2}$$

Here, $\mathbf{R}$ and $\mathbf{S}$ are the following 3×3 matrices:

$$\mathbf{R}=\begin{bmatrix}\mathbf{u}_k^{\mathsf{T}}\\ \mathbf{v}_k^{\mathsf{T}}\\ \mathbf{w}_k^{\mathsf{T}}\end{bmatrix} \quad\text{and}\quad \mathbf{S}=\begin{bmatrix}\frac{1}{\sigma_u^2} & 0 & 0\\ 0 & \frac{1}{\sigma_v^2} & 0\\ 0 & 0 & \frac{1}{\sigma_w^2}\end{bmatrix}, \tag{3}$$

where the unit column vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ are the dip and strike vectors of the kth nearby fault sample, respectively. The vector $\mathbf{w}_k=\mathbf{u}_k\times\mathbf{v}_k$ is normal to the plane of the kth fault sample, and σ_u, σ_v, and σ_w are the specified half-widths of the Gaussian function in the dip $\mathbf{u}$, strike $\mathbf{v}$, and normal $\mathbf{w}$ directions, respectively. The matrix $\mathbf{R}$ rotates the anisotropic Gaussian to be aligned with the vectors $\mathbf{u}_k$, $\mathbf{v}_k$, and $\mathbf{w}_k$. Because a fault should be locally planar in the strike and dip directions, we set the half-widths σ_u and σ_v to be larger than σ_w so that the Gaussian to be accumulated extends primarily in the fault strike and dip directions. For all examples in this paper, we set $\sigma_w=1$ and $\sigma_u=\sigma_v=15$ samples. The search window size used above is defined, from the maximum half-width of the anisotropic Gaussian, as $31=2\times15+1$.

To create fault samples located on ridges of a fault likelihood image, we must also construct images of fault strikes and dips. Therefore, when accumulating anisotropic Gaussian functions for the ith sample in the $5\times5\times5$ cube, we also accumulate weighted outer products of normal vectors for that sample as follows:

$$\mathbf{D}(\mathbf{x}_i)=\sum_{k=1}^{N} f(\mathbf{x}_k)g(\mathbf{x}_k-\mathbf{x}_i)\mathbf{w}_k\mathbf{w}_k^{\mathsf{T}}. \tag{4}$$

We then apply eigendecomposition to the 3×3 matrix $\mathbf{D}(\mathbf{x}_i)$, and we choose the eigenvector corresponding to the largest eigenvalue to be the normal vector $\mathbf{w}_i$ for the ith sample in the $5\times5\times5$ cube. From each normal vector $\mathbf{w}_i$, we then compute the strike and dip angles for the ith sample.

After constructing the three $5\times5\times5$ fault images of the fault likelihood, strike, and dip, we then create fault samples on the ridges of the fault likelihood image, and we search for missing neighbors among these new fault samples. Again, the conditions discussed in the previous section must be satisfied when finding neighbors from these new samples. If no valid neighbors can be found, we stop linking neighbors to the center fault sample.

Figure 7c shows new fault samples created in this way, colored yellow and blue for the two different fault surfaces. Using the newly created and original fault samples, we are able to construct intersecting fault surfaces without holes, as shown in Figure 7c.

Figure 6b shows the four fault surfaces extracted from the 3D seismic image by using the method discussed above. They can be displayed as opaque fault surfaces, as in Figure 6c, by simply increasing the size of each square so that they overlap and appear to form continuous surfaces.

However, these surfaces are really just linked fault samples. As shown in Figure 7c, the samples on a surface are linked above and below in the fault dip direction and left and right in the strike direction, and no holes are apparent. These links enable us to iterate among seismic image samples adjacent to a fault, in the dip and strike directions, as we estimate fault slips.

Recall that each sample in a fault surface corresponds to exactly one sample of the seismic image; therefore, fault likelihoods for the surfaces shown in Figure 6c can be easily displayed as a 3D fault likelihood image (with nonzero values only at faults) overlaid with the seismic image in Figure 8a. Compared to the thinned fault likelihood image in Figure 5a, spurious fault samples have been removed, and new fault samples have been created near the intersection (the dashed white circle) of faults F-A and F-B.

We create this fault likelihood image to constrain a structure-oriented filter (Fehmers and Höcker, 2003; Hale, 2009) so that it smooths along structures, but not across faults, to obtain the smoothed seismic image shown in Figure 8b. We use this smoothed image in the next section for estimating fault slips because the smoothing does what seismic interpreters do visually when estimating fault slips, by bringing seismic amplitudes from within each fault block up to, but not across, the faults.

FAULT DIP SLIPS

In a 3D seismic image, fault strike slips are typically less apparent than dip slips. Therefore, we have not attempted to estimate fault slips in the strike direction. As shown in Figure 2a, fault dip slip is a vector representing displacement, in the dip direction, of the hanging wall side of a fault surface relative to the footwall side. Fault throw is the vertical component of slip. If we know the fault throw and the fault surface with linked samples, as in Figure 7c, then we can walk up or down the fault in the dip direction to compute the two corresponding horizontal components of dip slip. Therefore, to estimate dip slips, we first estimate fault throws.

Fault throws

To estimate fault throws, we compute vertical components of shifts that correlate seismic reflectors on the footwall and hanging wall sides of a fault surface. As discussed by Hale (2013b), this correlation can be difficult. The drawing of a fault in Figure 2a is a very simple case, in which the fault surface is entirely planar and the fault throw is constant for the entire surface. In reality, fault throws may vary significantly within a fault surface, and this variation can make windowed crosscorrelation methods (Aurnhammer and Tonnies, 2005; Liang et al., 2010) fail when fault throws vary within a chosen window size.

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

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

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

In this paper, we use the same dynamic warping process, but with seismic amplitude differences computed using the simpler linked data structure illustrated in Figure 7c. It is advantageous that the fault surfaces represented in Figure 7c do not have holes. Holes

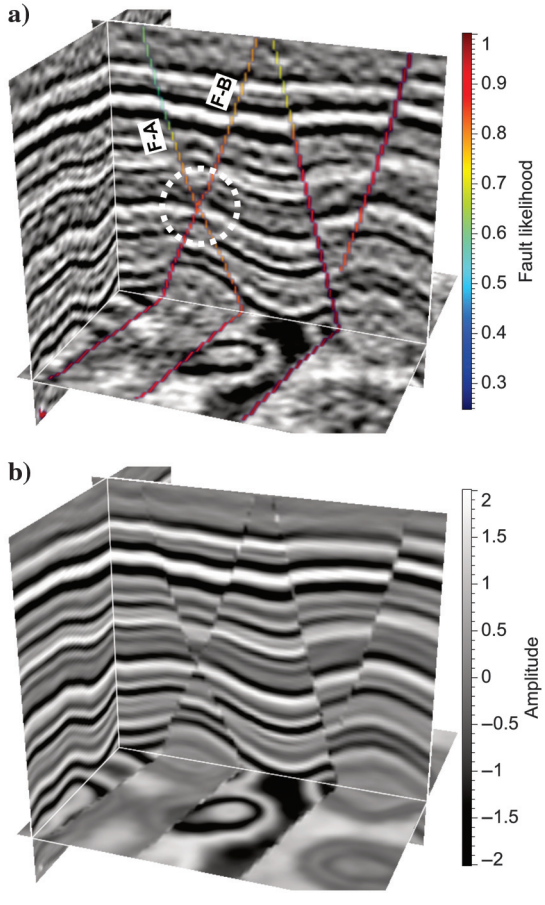

Figure 8. Linked fault samples in Figure 6c can be displayed as a fault likelihood image (with mostly zeros) overlaid with the seismic image in panel (a). Compared to the thinned fault likelihood image in Figure 5a, spurious fault samples have been removed. New fault samples are created at the intersection (dashed white circle) of faults A and B when constructing surfaces. This fault image is used to constrain a structure-oriented smoothing filter so that it smooths the seismic image along structures but not across the faults as shown in panel (b).

in fault surfaces such as those shown in Figure 7b make it difficult to determine which samples of the seismic image should be used when computing the amplitude differences required for dynamic warping. Holes also make it difficult for the dynamic warping method to enforce the constraints that fault throws vary smoothly. For these reasons, fault throws estimated using fault surfaces such as those shown in Figure 7c are more accurate than those for fault surfaces with holes, such as those shown in Figure 7b.

The fault surfaces in Figure 9a are the same surfaces shown in Figure 6c, but they are colored by fault throws estimated using the method discussed above. We observe that fault throws estimated for each surface vary smoothly, as expected. In addition, fault throws for fault F-C are negative because this fault is a reverse fault. Estimated fault throws for faults F-A, F-B, and F-C generally increase in magnitude with depth, whereas throws for the smaller fault F-D first increase, then decrease with depth.

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

If the computed fault dip slip vectors are accurate, then we should be able to undo faulting apparent in the seismic image, that is, to correctly align seismic reflectors on opposite sides of faults. An unfaulting process described below verifies the accuracy of these estimated fault dip slip vectors.

Unfaulted images

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

Here, we use a different method described by Wu et al. (2015) to unfault a seismic image, and we thereby verify our estimated dip slips. In this method, vector shifts are computed for all samples in the seismic image by solving the partial differential equations derived from the fault slip vectors estimated at faults. This method moves the footwalls and hanging walls and even faults themselves simultaneously, to undo faulting with minimal distortion.

Figure 9b shows the seismic image with faults colored by estimated fault throws, the vertical components of estimated dip slip vectors. These fault throws and the two horizontal components of the slips, are used by Wu et al. (2015) to obtain the unfaulted image shown in Figure 9c. In the unfaulted image, seismic reflectors are well aligned across faults, including the intersecting normal faults and the reverse fault. This unfaulted image illustrates that the estimated fault slip vectors are accurate to within the resolution of the seismic image.

A REAL IMAGE EXAMPLE

The synthetic 3D seismic image shown in Figures 3–9 illustrates our 3D seismic image processing for (1) computing fault images, (2) constructing fault samples, (3) constructing fault surfaces, (4) estimating fault dip slips, and (5) unfaulting the seismic image to assess the accuracy of those slip vectors. This synthetic example also demonstrates that this image processing works for normal faults, reverse faults, and fault intersection.

A real 3D seismic image, from an onshore 3D seismic survey over the Schoonebeek oilfield, is used here as a further demonstration of the same processing. In this real seismic image shown in Figure 10 (and in the smaller subset shown in Figure 1), many faults are apparent and many of them intersect with others. With such faults, this image is a good example and summary of the methods discussed above.

1) From this 3D seismic image, images of fault likelihood, strike, and dip are first computed by scanning over a range of possible

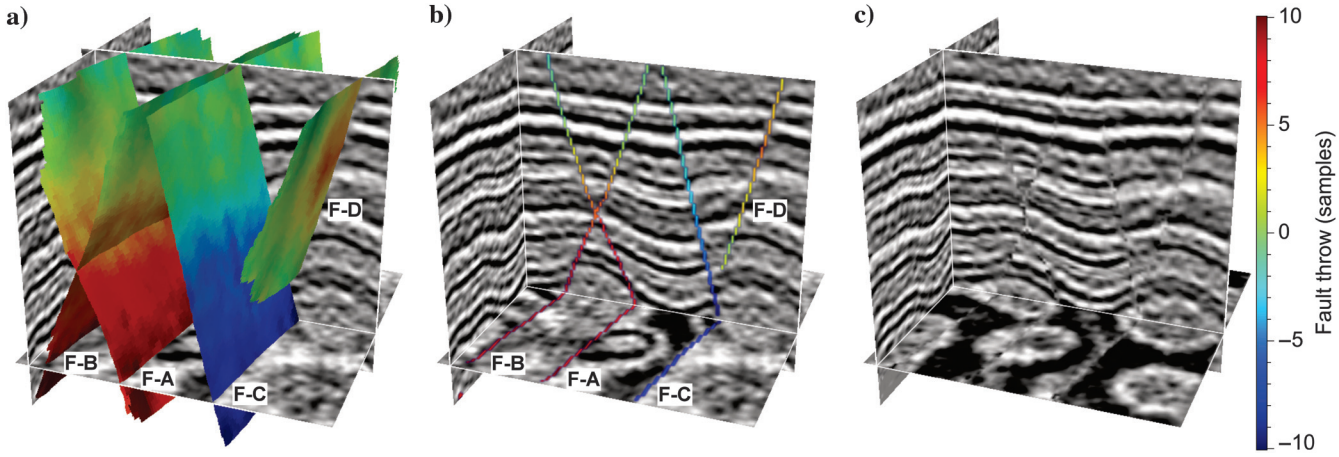

Figure 9. (a) Fault surfaces and fault throws for a 3D seismic image (b) before and (c) after unfaulting. In all image slices, reflectors are more continuous after unfaulting.

strikes and dips with a semblance-based filter that highlights locally planar discontinuities.

2) These three fault images are then represented by fault samples, which are displayed as squares oriented by strikes and dips and colored by fault likelihood in the upper right panel of Figure 10a. Remember that each fault sample corresponds to a seismic image sample in the sampling grid of the seismic image; therefore, the same fault samples can be displayed as a fault likelihood image overlaid with the seismic image slices shown in Figure 10a.
3) The oriented fault samples are then linked to form fault surfaces, displayed in the upper right panel of Figure 10b. Many of these fault surfaces intersect each other, and the differences in strikes for these intersecting faults are approximately 60°. These fault surfaces are really just sets of linked fault samples located within the sampling grid of the seismic image; they appear as surfaces only because the squares representing the fault samples are displayed with sizes large enough to overlap with each other. These linked fault samples can also be displayed as a fault likelihood image overlaid with the seismic image in Figure 10b. In the constant-time slice, we observe complicated intersections among the extracted fault surfaces.

   Compared to the three slices in Figure 10a, some fault samples are removed when constructing surfaces because they cannot be linked to form surfaces with significant sizes. In this example, we discarded fault surfaces with fewer than 2000 samples. In addition, new fault samples are created to fill holes that occur where faults intersect.
4) These fault surfaces of linked fault samples are further used to estimate fault dip slips. Fault throws (vertical component of slips) are displayed on fault surfaces in the upper right panel of Figure 11. After estimating fault slips, the number of fault surfaces is reduced because we keep only fault surfaces for which dip slips are significant. Again, each fault sample in a fault surface corresponds to exactly one sample of the 3D seismic image, so fault throws can be displayed as a 3D image overlaid with the seismic image as in Figure 11a.
5) Using the estimated fault dip slip vectors, the seismic image can be unfaulted as shown in Figure 11b. In the unfaulted image,

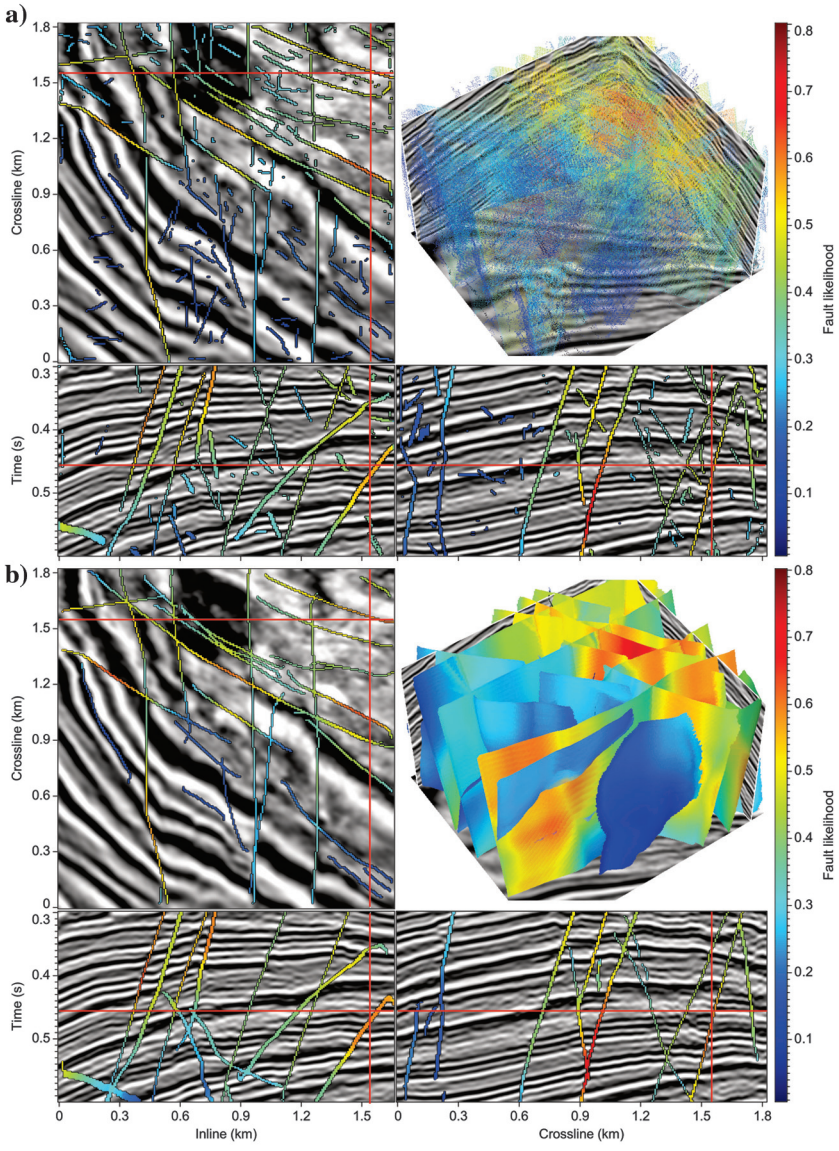

Figure 10. Fault samples colored by fault likelihood (a) are computed and (b) linked to form fault surfaces.

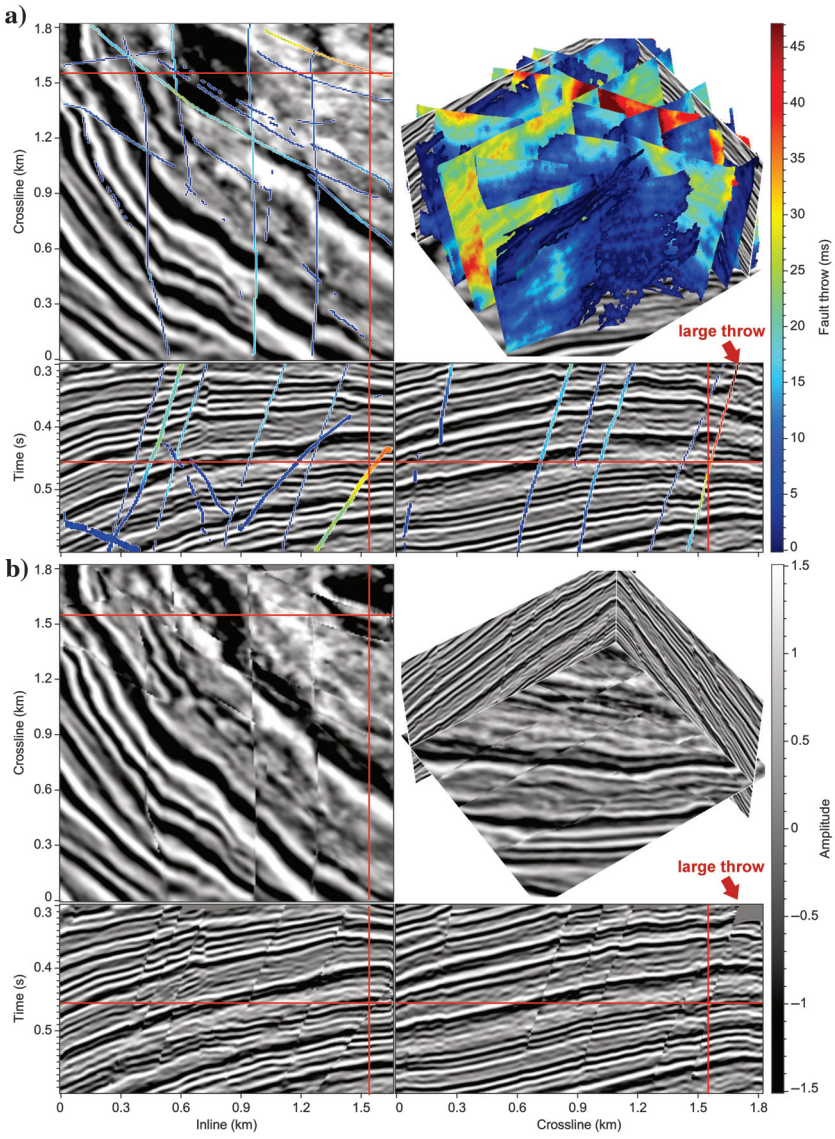

Figure 11. Fault surfaces and fault throws for a 3D seismic image (a) before and (b) after unfaulting. In all image slices, reflectors are more continuous after unfaulting. The red arrows point to a large-slip fault before and after unfaulting.

seismic reflectors in all image slices are more continuous than those in the original image slices shown in Figure 11a. For the fault with large slips highlighted by the red arrow in Figure 11a, footwall and hanging wall sides are moved significantly to align the reflectors on these opposite sides, as shown in Figure 11b.

CONCLUSIONS

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

Linked fault samples can also be displayed as fault surfaces by simply increasing sizes of squares used to represent fault samples. These fault surfaces, however, are not triangular or quadrilateral meshes, which are unnecessarily complicated for our processing. This simple linked data structure is easy to construct and convenient to exchange between computers. This structure also facilitates subsequent image processing for faults, such as fault slip estimation and seismic image unfaulting.

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

Most of the computation time in the whole image processing is spent in the first step for computing the fault likelihood, strike, and dip images. For the 3D real example with $200 \times 208 \times 228$ image samples, the method requires approximately 20 min to compute all the three fault images. Constructing fault surfaces from fault samples is costless if the fault geometry is simple and only a small number of fault samples needed to be interpolated to fill holes.

ACKNOWLEDGMENTS

This research is supported by the sponsors of the Consortium Project on Seismic Inverse Methods for Complex Structures. We appreciate suggestions by A. E. Barnes, K. J. Marfurt, R. Heggland, and one anonymous reviewer that led to significant revision of this paper. The real 3D seismic image used in this paper was graciously provided by K. Rutten and B. Howard, via the Netherlands Organisation for Applied Scientific Research.

REFERENCES

Admasu, F., 2008, A stochastic method for automated matching of horizons across a fault in 3D seismic data: Ph.D. thesis, Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek.

Admasu, F., S. Back, and K. Toennies, 2006, Autotracking of faults on 3D seismic data: Geophysics, **71**, no. 6, A49–A53, doi: 10.1190/1.2358399.

Aqrawi, A. A., and T. H. Boe, 2011, Improved fault segmentation using a dip guided and modified 3D Sobel filter: 81st Annual International Meeting, SEG, Expanded Abstracts, 999–1003.

Aurnhammer, M., and K. Tonnies, 2005, A genetic algorithm for automated horizon correlation across faults in seismic images: IEEE Transactions on Evolutionary Computation, **9**, 201–210, doi: 10.1109/TEVC.2004.841307.

Borgos, H. G., T. Skov, T. Randen, and L. Sønneland, 2003, Automated geometry extraction from 3D seismic data: 73rd Annual International Meeting, SEG, Expanded Abstracts, 1541–1544.

Fehmers, G. C., and C. F. Höcker, 2003, Fast structural interpretation with structure-oriented filtering: Geophysics, **68**, 1286–1293, doi: 10.1190/1.1598121.
Gibson, D., M. Spann, J. Turner, and T. Wright, 2005, Fault surface detection in 3-D seismic data: IEEE Transactions on Geoscience and Remote Sensing, **43**, 2094–2102, doi: 10.1109/TGRS.2005.852769.
Hale, D., 2009, Structure-oriented smoothing and semblance: CWP Report 635.
Hale, D., 2013a, Dynamic warping of seismic images: Geophysics, **78**, no. 2, S105–S115, doi: 10.1190/geo2012-0327.1.
Hale, D., 2013b, Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3D seismic images: Geophysics, **78**, no. 2, O33–O43, doi: 10.1190/geo2012-0331.1.
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.
Liang, L., D. Hale, and M. Maučec, 2010, Estimating fault displacements in seismic images: 80th Annual International Meeting, SEG, Expanded Abstracts, 1357–1361.
Luo, S., and D. Hale, 2013, Unfaulting and unfolding 3D seismic images: Geophysics, **78**, no. 4, O45–O56, doi: 10.1190/geo2012-0350.1.
Marfurt, K. J., R. L. Kirlin, S. L. Farmer, and M. S. Bahorich, 1998, 3-D seismic attributes using a semblance-based coherency algorithm: Geophysics, **63**, 1150–1165, doi: 10.1190/1.1444415.
Marfurt, K. J., V. Sudhaker, A. Gersztenkorn, K. D. Crawford, and S. E. Nissen, 1999, Coherency calculations in the presence of structural dip: Geophysics, **64**, 104–111, doi: 10.1190/1.1444508.
Pedersen, S. I., T. Randen, L. Sonneland, and Ø. Steen, 2002, Automatic fault extraction using artificial ants: 72nd Annual International Meeting, SEG, Expanded Abstracts, 512–515.
Pedersen, S. I., T. Skov, A. Hetlelid, P. Fayemendy, T. Randen, and L. Sønneland, 2003, New paradigm of fault interpretation: 73rd Annual International Meeting, SEG, Expanded Abstracts, 350–353.
Randen, T., S. I. Pedersen, and L. Sønneland, 2001, Automatic extraction of fault surfaces from three-dimensional seismic data: 81st Annual International Meeting, SEG, Expanded Abstracts, 551–554.
Van Bemmel, P. P., and R. E. Pepper, 2000, Seismic signal processing method and apparatus for generating a cube of variance values: U. S. Patent 6,151,555.
Walsh, J., J. Watterson, W. Bailey, and C. Childs, 1999, Fault relays, bends and branch-lines: Journal of Structural Geology, **21**, 1019–1026, doi: 10.1016/S0191-8141(99)00026-7.
Wu, X., S. Luo, and D. Hale, 2015, Moving faults while unfaulting 3D seismic images: 85th Annual International Meeting, SEG, Expanded Abstracts, 1692–1697.