3D seismic image processing for faults

Xinning Wu & Dave Hale
Center for Wave Phenomena, Colorado School of Mines, Golden, CO 80401, USA

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

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

Key words: seismic faults image processing

1 INTRODUCTION
Faults like 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 are also proposed to extract fault surfaces from computed fault images, Pedersen et al. (2002) and Pedersen et al. (2003) propose the ant tracking method to first extract small fault segments, which are then merged to form larger fault surfaces. Similarly, the methods, proposed by Gibson et al. (2005), Admasu et al. (2006) and Kadlec et al. (2008), also try to build larger fault surfaces from smaller patches. Hale (2013b) uses images of fault likelihoods, strikes and dips to construct fault surfaces that coincide with ridges of the fault likelihood image.
Figure 2. Fault dip slip (a) is a vector representing displacement, in the dip direction, of the hanging wall side of a fault surface relative to the footwall side. Fault throw is the vertical component of the slip. Fault strike and dip angles, with corresponding unit vectors are defined in (b).

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

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

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

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

We then use the fault samples in Figure 1b to construct fault surfaces, which appear to be continuous, as shown in Figure 1c, but are actually only linked lists 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.
Figure 4. A 3D seismic image with fault likelihoods (a), strikes (b) and dips (c) displayed in color. The dashed white circle in each image indicates one location where fault F-A intersects fault F-B.

Figure 5. Thinned fault likelihood image (a) has non-zero 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 (b) and (c), respectively. The dashed white circle in each image indicates one location where fault F-A intersects fault F-B.
2 FAULT IMAGES

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

In 3D seismic images like those shown in Figures 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 fault likelihood (Figure 4a), while at the same time estimating fault strike (Figure 4b) and dip (Figure 4c). The fault likelihood image indicates where faults might exist, while the strike and dip images indicate their orientations.

In computing these fault images, this method scans over a range of possible combinations of strike and dip to find the one orientation that maximizes the fault likelihood, for each image sample (Hale, 2013b). The maximum fault likelihood for each image sample is recorded in the fault likelihood image (Figure 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 where faults might exist. In the fault strike (Figure 4b) and dip (Figure 4c) images, we expect strike and dip angles to be accurate only where faults are likely, that is, where fault likelihoods are high.

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

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

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

3 FAULT SAMPLES AND SURFACES

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

3.1 Fault samples

Because most samples in the images of fault likelihood (Figure 5a), strike (Figure 5b) and dip (Figure 5c) are null, 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 fault likelihood, while the orientation of each square represents fault strike and dip. Fault samples exist only at positions where thinned fault likelihoods are non-null, and each fault sample corresponds to one and only one image sample. Therefore, these fault samples contain exactly the same information represented in the thinned fault images shown in Figure 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 6 and 7. These misaligned samples, however, cannot be linked together to form locally planar fault surfaces of significant size.
Figure 6. The three fault images in Figure 5 can be represented by fault samples displayed as small squares (a). Each square in (a) is colored by fault likelihood and oriented by the strike and dip of the corresponding fault sample. Links are then built among consistent fault samples, and each set of linked fault samples in (b) represents a fault surface that appears opaque in (c), where fault samples are displayed as larger overlapping squares.

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

3.2 Fault surfaces

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

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

3.2.1 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 sufficiently high fault likelihood. Remember that each fault sample corresponds to exactly one sample of the seismic image. This means that we can use the image sampling grid to efficiently search for neighbor samples that should be linked. In a 3D sampling grid, each fault sample has 26 adjacent grid points in a $3 \times 3 \times 3$ cube centered at that sample, but most of these adjacent grid
points will not have a fault sample. At these adjacent grid points, we search for up to four neighbor fault samples, above and below (in directions best aligned with fault dip), left and right (in directions best aligned with fault strike).

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

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

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

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

Some samples linked in this way may not correspond to faults. We discard surfaces with small numbers of linked samples, and keep only those with significant sizes. For an example, in Figure 7b, we have kept only the three largest surfaces constructed from the fault samples in Figure 7a. Other fault samples (such as these colored by green and blue in Figure 7a) are then ignored in subsequent processing.

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 the seismic image is noisy. And 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, like the fault F-B shown in Figure 7b, and can yield gaps which 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.

### 3.2.2 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 more complete fault surface. Instead, we check for missing neighbors, and create them as we grow fault surfaces, and thereby directly obtain complete fault surfaces without holes as shown in Figure 7c.

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

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

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

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

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

\[
g(x) = \exp(-\frac{1}{2} x^T R^T S R x).
\]
Here, $\mathbf{R}$ and $\mathbf{S}$ are $3 \times 3$ matrices:

$$
\mathbf{R} = \begin{bmatrix}
    u_k^\top \\
    v_k^\top \\
    w_k^\top
\end{bmatrix}, \quad \text{and,} \quad \mathbf{S} = \begin{bmatrix}
    \frac{1}{\sigma_u} & 0 & 0 \\
    0 & \frac{1}{\sigma_v} & 0 \\
    0 & 0 & \frac{1}{\sigma_w}
\end{bmatrix}, \quad (3)
$$

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

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

$$
\mathbf{D}(x_i) = \sum_{k=1}^{N} f(x_k) g(x_k - x_i) w_k w_k^\top. \quad (4)
$$

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

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

Figure 7c shows new fault samples created in this way, and colored by yellow and blue, for two different fault surfaces. Using both newly created and original fault samples, we are able to construct intersecting fault surfaces without holes, as shown in Figure 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, left and right in the strike direction, and no holes are apparent. These links enable us to iterate among seismic image samples adjacent to a fault, in both dip and strike directions, as we estimate fault slips.

Recall that each sample in a fault surface corresponds to exactly one sample of the seismic image, therefore, fault likelihoods for the surfaces shown in Figure 6c can be easily displayed as a 3D fault likelihood image (with non-null values only at faults) overlayed with the seismic image in (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. The seismic image in (b) is smoothed along structures, but not across the faults.
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.

4 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.

4.1 Fault throws

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

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

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

Hale (2013b) represents a fault surface as a quad mesh that facilitates computation of differences between seismic amplitudes on opposite sides of a fault surface. As discussed by Hale (2013b), this correlation can be difficult. The drawing of a fault in Figure 2a is a very simple case, where the fault surface is entirely planar, and fault throw is constant for the entire surface. In reality, fault throws may vary significantly within a fault surface, and this variation can make windowed cross-correlation methods (Aurnhammer and Tonnies, 2005; Liang et al., 2010) fail when fault throws vary within a chosen window size.
with depth. An unfaulting processing described below verifies the accuracy of these estimated fault throws.

Fault surfaces in Figure 9a are the same surfaces shown in Figure 6c, but colored by fault throws estimated using the method discussed above. We observe that fault throws estimated for each surface vary smoothly, as expected. Also, fault throws for fault F-C are negative because this fault is a reverse fault. Estimated fault throws for faults F-A, F-B, and F-C generally increase in magnitude with depth, while throws for the smaller fault F-D first increase, then decrease with depth. An unfaulting processing described below verifies the accuracy of these estimated fault throws.

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

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

### 4.2 Unfaulted images

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

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

Figure 9b shows the seismic image with faults colored by estimated fault throws, the vertical components of estimated dip slip vectors. These fault throws, as well as the two horizontal components of the slips, are used in Wu et al. (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 estimated fault slip vectors are accurate to within the resolution of the seismic image.

### 5 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 and (3) fault surfaces, (4) estimating fault dip slips, and (5) unfaulting the seismic image to assess the accuracy of those slip vectors. This synthetic example also demonstrates that this image processing works for both normal and reverse faults, and for intersecting faults.

A subset of a real 3D seismic image, provided by Kees Rutten and Bob Howard via TNO, is used here as a further demonstration of the same processing. In this real seismic image shown in Figure 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 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 right-upper 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 overlayed with the seismic image slices shown in Figure 10a.

3. The oriented fault samples are then linked to form fault surfaces, displayed in the right-upper panel of Figure 10b. Many of these fault surfaces intersect each other, and the differences in strikes for these intersecting faults are approximately 60 degrees. These fault surfaces are really just linked lists of fault samples located within the sampling grid of the seismic image; they appear as surfaces only because the squares representing the fault samples are displayed with sizes large enough to overlap.
Figure 10. Fault samples colored by fault likelihood (a) are computed, and linked to form fault surfaces (b).
Figure 11. Fault surfaces and fault throws for a 3D seismic image before (a) and after (b) unfaulting. In all image slices, reflectors are more continuous after unfaulting. The red arrows point to a large-slip fault before and after unfaulting.
with each other. These linked fault samples can also be displayed as a fault likelihood image overlayed 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. Also, new fault samples are created to fill holes that occur where faults intersect.

(4) These fault surfaces of linked fault samples are further used to estimate fault dip slips. Fault throws (vertical component of slips) are displayed on fault surfaces in the upper-right panel of Figure 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 overlayed 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, 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.

6 CONCLUSION

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

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

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

ACKNOWLEDGMENTS

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

REFERENCES


———, 2013b, Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3D seismic images: Geophysics, 78, O33–O43.


