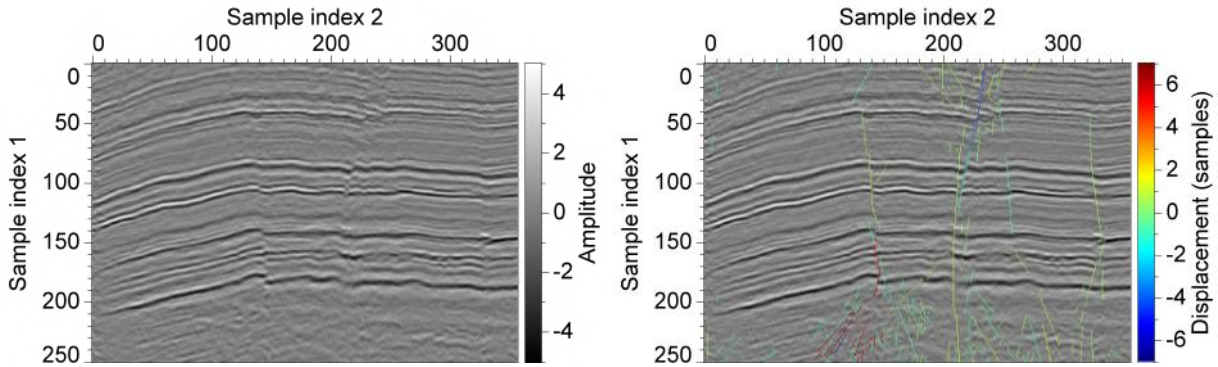
Estimating fault displacements in seismic images

Luming Liang[1], Dave Hale[1] & Marko Maučec[2]

[1] *Center for Wave Phenomena, Colorado School of Mines, Golden, CO 80401, USA*

[2] *Landmark Graphics Corporation, Halliburton, Highlands Ranch, CO 80129, USA*



(a) (b)

Figure 1. A 2D seismic section (a) showing extensional faulting and the same section with vertical displacements of faults highlighted (b). Signs of displacements indicate that geological layers have been displaced either downward (positive) or upward (negative) from left to right.

ABSTRACT

Geologic faults complicate the mapping of depositional layers. Most existing seismic image processing techniques highlight *fault locations* but fail to estimate *fault displacements*.

We model faults as a displacement vector field. Unlike traditional attributes (e.g., semblance or coherence), our estimated fault displacement vector field provides information about fault displacements, as well as fault locations. This vector field can be used to automatically determine relative displacements of faulted layers, and thereby simplify the mapping of such layers.

Key words: fault displacements, crosscorrelation, seismic interpretation

1 INTRODUCTION

In seismic images of the earth's subsurface, such as the one shown in Figure 1a, we typically see interfaces between geologic layers, because rocks or fluids within these layers vary from one layer to the next, causing changes in acoustic impedance. It is these changes that we image with seismic waves.

Discontinuities are often apparent in these images. In Figure 1a, the two most obvious discontinuities appear in the central part of the image. Discontinuities of this kind correspond to geologic faults, which are fractures in rocks. Faults tend to be more vertical than layers, as rocks on one side of a fault tend to be displaced downward or upward relative to rocks on the other side. In seismic images, faults appear as discontinuities in otherwise nearly continuous layers. Geoscientists quickly learn to estimate the amount of displacement along a fault as they attempt to unravel the geologic history of the subsurface that has been imaged.

In an ideal 2D seismic image, without any random

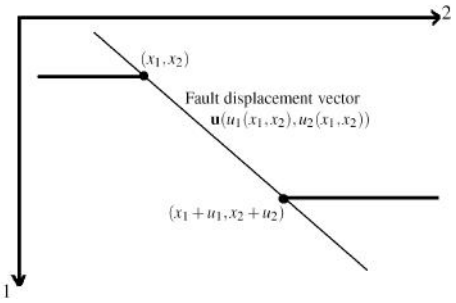


Figure 2. The fault displacement vector $u(x)$.

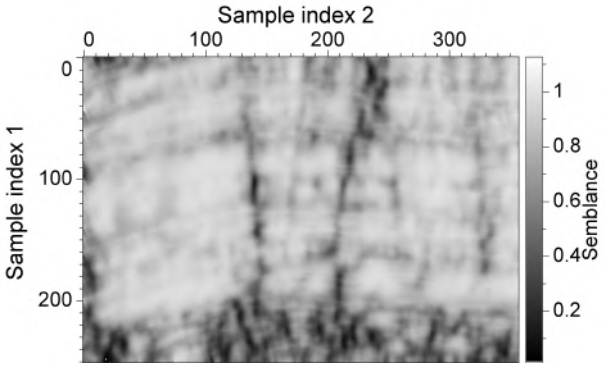


Figure 3. Semblance for the seismic image in Figure 1a, computed using the method described by Hale (2009b).

or coherent noise, faults appear as curves of discontinuities. These discontinuities are easily detected by human interpreters. Because faults never consist of a single fracture the term "fault zone" is sometimes used (Kadlec et al., 2008). However, in this paper, we consider a fault to be a single curve.

Seismic image processing today routinely includes steps to highlight the locations of faults (Bahorich and Farmer, 1995; Cohen and Coifman, 2002; Gibson et al., 2005; Al-Dossary and Marfurt, 2006; Kadlec et al., 2008; Hale, 2009b). That processing cannot yet reliably estimate the displacements along the faults. Fault displacements are today estimated manually using a tedious process of viewing seismic images and interactively picking corresponding points on both sides of a fault.

We seek to replace this manual picking with a process that produces an image of fault displacements. This process extracts fault curves from 2D seismic images and estimates fault displacements along these curves.

1.1 Fault model

Many mathematical models for faults have been proposed by geologists and geophysicists (Watterson, 1986; Barnett et al., 1987) based on their investigations and conclusions about seismic images and well logs. Barnett et al. (1987) claim that fault displacements consist of two components which can be considered separately: near-field and far-field. The near-field components are displacements that occur in the rock volume closely surrounding a fault. The far-field components are related to the bulk deformation of a larger region. Our model considers only near-field displacements.

In this paper, we consider only 2D slices of 3D seismic images, so that faults appear as curves of discontinuities in a 2D seismic image. Ideally, faults we detect are exactly one pixel wide, see Figure 1b.

Fault displacement vectors, which have both vertical and horizontal components, vary continuously along these curves. We represent the seismic image with a 2D scalar array: $f(x_1, x_2)$, where x_1 and x_2 denote uniformly sampled vertical and horizontal coordinates, respectively. An estimated fault displacement is a 2D vector: $\mathbf{u} = \mathbf{u}(u_1, u_2)$ (as in Figure 2), where $u_1 = u_1(x_1, x_2)$ and $u_2 = u_2(x_1, x_2)$ are vertical and horizontal components of the displacement, respectively. In Figure 1b, colored pixels represent the vertical component u_1 of the vectors $\mathbf{u}$. Those vectors should be zero (or nearly zero) at locations where no faults are present.

A nearly horizontal fault cannot be easily detected, even by human interpreters. Consequently, faults considered here are more vertical than horizontal; we assume that the angle between a fault curve and a vertical line is less than 45 degrees.

1.2 Estimating fault location

A feature shared by most fault detection techniques is the computation of attributes that highlight discontinuities in seismic images. Examples of such attributes include coherence (Bahorich and Farmer, 1995), entropy (Cohen and Coifman, 2002), curvature (Al-Dossary and Marfurt, 2006) and semblance (Hale, 2009b). Depending on which attribute is chosen, discontinuities in seismic images cause attribute values to be anomalously low or high.

Figure 3 illustrates structure-oriented semblance computed using the method by Hale (2009b). Potential fault locations are indicated by dark pixels. However, the information we obtain from this attribute is limited. Darker areas indicate fault zones but fail to locate fault curves. Fault locations are poorly resolved. This defect is common in this sort of algorithm.

This resolution problem is often due to the use of overlapping windows of samples when computing seismic attributes. Windows containing imaged faults smear information from both sides of the faults, over a distance proportional to the effective window width. Smaller windows cause less smearing, but yield less accurate measurements of discontinuity.

To overcome this problem, methods that involve more heuristics or human interactions have been proposed. Kadlec et al. (2007) present an interactive method for computing a fault surface. Rather than us-

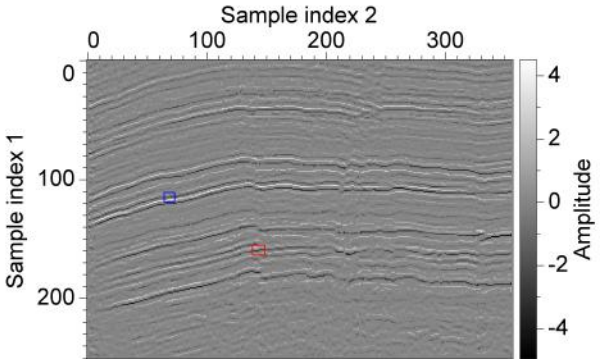


Figure 4. Automatic horizon picking using ridges and valleys detection as described by Patel et al. (2008). No significant differences exist between a fault area (red box) and an unfaulted area (blue box).

ing the attributes mentioned above, they manually pick seed points in the image and then let the points evolve to surfaces using a level set method.

Gibson et al. (2005) present a similar method. They also model faults in 3D as growing surfaces, but require no human interaction, because the seed points are generated from semblance. Then, these points are grouped into small fault patches using a highest-confidence-first merging strategy.

These methods compare automatic fault detection with human picking. The authors claim advantages in automatic fault detection over human picking in that the methods save time and are more accurate.

1.3 Research on fault displacement

Compared to automatic fault picking, automatic estimation of fault displacements is potentially more useful but less well developed. As mentioned above, fault displacements are today usually estimated manually by experienced interpreters.

Methods for automatically picking horizons in seismic images are most closely related to our work. Techniques of this kind appear also in the fields of computer vision and machine intelligence, although the problem of how to automatically pick horizons stems from geophysical applications.

In the method described by Farakloioti and Petrou (2004), horizons can be automatically identified as median surfaces of layers. The key part of this process is connected-component analysis, which is designed to join fragments that have consistent orientation and proximity. However, this method does not attempt to find correspondences between horizons on each side of a fault and leaves fault zones blank.

New approaches that use genetic algorithms (Aurnhammer and Tönnies 2005) and a multi-scale Bayesian model (Admasu and Tönnies, 2006) have been proposed to solve this problem. The former algorithm works for only 2D seismic images; the latter works for both 2D and 3D images. In addition to seismic images, these algorithms require horizon curves or surfaces in unfaulted regions as additional information; e.g., median surfaces extracted by Farakloioti and Petrou (2004) or valleys and ridges traced out by Patel et al. (2009). These two approaches match layers on both sides of faults, but neither yields displacements along fault curves or surfaces. Moreover, both approaches are sensitive to errors in the input horizons. This sensitivity is illustrated in Figure 4. In the case shown in Figure 4, two layers on each side of a fault join coincidentally in the red box. The method mistakenly picked one horizon across the fault in the red box. Because the method cannot distinguish the faulted and unfaulted areas highlighted there, layers would be mistakenly matched.

For these reasons, newer algorithms are based on interactive horizon picking. Patel et al. (2008; 2009) propose a framework for computer-assisted seismic analysis, designed for a small group of interpreters. Their first step is automatic horizon picking in unfaulted areas, which is similar to Farakloioti and Petrou's (2004) method. By considering the amplitude of a 2D seismic image as height values in a terrain, Patel et al. trace out the valleys and ridges, which are texture primitives defined by Tüceryan and Jain (1990). Then, they create curves from these valleys and ridges as horizons. The difference is in their second step. Rather than relying on an entirely automatic method, they employ a semi-automatic method that uses human intervention when the auto-picking result is incorrect, as in the red square in Figure 4.

In this paper, we describe a method for automatically and simultaneously estimating both fault locations and fault displacements. We require only a seismic image as the input. Faults are simply located where our estimated fault displacements are nonzero.

2 PROBLEMS

A simple way to estimate vertical displacements across faults is to crosscorrelate each pair of adjacent traces in a seismic image. We search for peaks of normalized local crosscorrelations to estimate displacements between traces. Local correlations enable us to estimate displacements that may vary vertically. Normalization makes our estimates insensitive to vertical variations in seismic amplitudes.

Normalized local crosscorrelations are often used in geophysical applications; e.g., to estimate relative displacements from two time-lapse seismic images (Hale, 2009a) and to enhance stacking (Liu et al., 2009).

Normalized local crosscorrelations are defined for

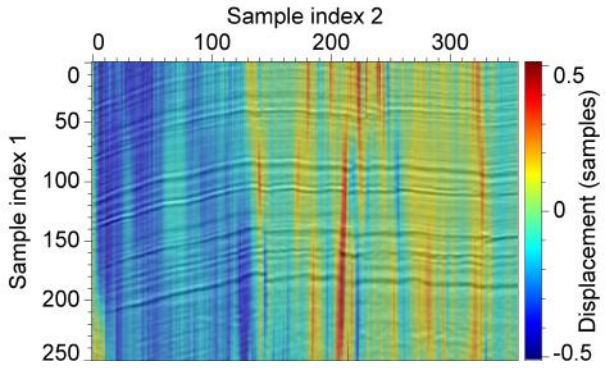


Figure 5. The estimated vertical displacements using the simple method.

two traces f and g by

$$c[k;l] \equiv \frac{c_{fg}[k;l]}{\sqrt{c_{ff}[k;0]}\sqrt{c_{gg}[k+l;0]}}, \quad (1)$$

where

$$c_{fg}[k;l] \equiv \sum_j f[j]g[j+l] \times w[k-j] \quad (2)$$

and $w[k]$ is a Gaussian window

$$w[k] \equiv e^{\frac{-k^2}{2\sigma^2}} \quad (3)$$

that, for some specified radius σ, makes crosscorrelations local.

Crosscorrelations are normalized by the factors

$$c_{ff}[k;0] \equiv \sum_j f^2[j] \times w[k-j] \quad (4)$$

and

$$c_{gg}[k;0] \equiv \sum_j g^2[j] \times w[k-j]. \quad (5)$$

For each integer lag l, equation 1 gives a normalized local crosscorrelation coefficient for every sample indexed by k. We use a fast implementation of equation 1 (Hale, 2006). When estimating displacements, we must store for each sample only those correlation coefficients required to locate correlation peaks. For example, if the maximum correlation coefficient appears at lag l, we need only store values for lags $l-1$, l, and $l+1$. We then fit a quadratic function to those values to locate the correlation peak with sub-pixel precision. Details are described by Hale (2009a), and source code is available in the Mines Java Toolkit (http://mines.edu/~dhale/jtk/). Figure 5 shows vertical displacements estimated by finding the peaks of local crosscorrelations of consecutive pairs of traces in the seismic section shown in Figure 1a.

Although these trace-to-trace vertical displacements roughly conform to the orientations of the geological layers in the seismic image, they do not properly describe fault displacements. Estimated displacements are apparently incorrect at some locations, such as the point near a fault with sample indices (160,143). Near this point, vertical displacements across the fault should be positive (about 5 or 6 samples), which means that the relative displacement of layers is downward from left to right. However, the estimated displacement value at that point is negative. Furthermore, faults appear as zones of displacement, not as curves, and displacements are not zero where layers are dipping but not faulted. In the remainder of this section, we describe in detail the shortcomings of this simple trace-to-trace correlation method for estimating fault displacements.

2.1 Inadequacy of using two traces

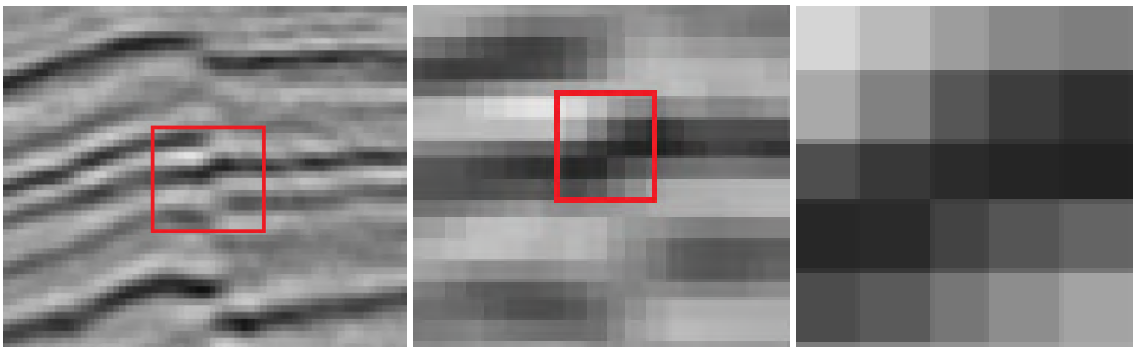
In Figure 6 are three subsets of pixels centered on the pixel with sample indices $(160, 143)$ in Figure 1a. At this location, two layers on each side of a fault join coincidentally. Human interpreters can estimate the correct relative displacement of layers on each side; however, the simple method fails to do so. One problem with the simple method is that it is near sighted, as illustrated in Figure 6.

Using our eyes, we can easily identify the location of the fault and roughly estimate the displacement between layers on each side of the fault shown in Figure 6a. From left to right across the fault, layers are clearly displaced downward.

However, if we zoom in the image, and only look at Figure 6b, we may misinterpret the fault displacements. In this case, we can still see the location of the fault, but may unfortunately pick a wrong correspondence between layers on each side of the fault. If we continue to magnify the image until we can see only the 5×5 region shown in Figure 6c, we may even be unaware of any faulting. Note that the window size used in many such image processing applications is often kept small because computational cost is higher if larger windows are used.

When computing local crosscorrelations for pairs of traces, our window height is controlled by the Gaussian half-width σ in equation 3, but the window width is only 2 samples. Correlating local windows of only two traces near a fault is insufficient and quite different from what human interpreters do. Human interpreters use traces farther away from a fault to both locate the fault and estimate displacements.

Of course, correlating a single pair of traces that are farther apart will introduce another source of error, as displacements estimated from traces away from a fault may poorly approximate those apparent at a fault. What human interpreters correlate visually are not single pairs of traces, but many traces on both sides of faults.



(a) (b) (c)

Figure 6. Zoomed views of the fault located around $(160, 143)$ in Figure 1a.

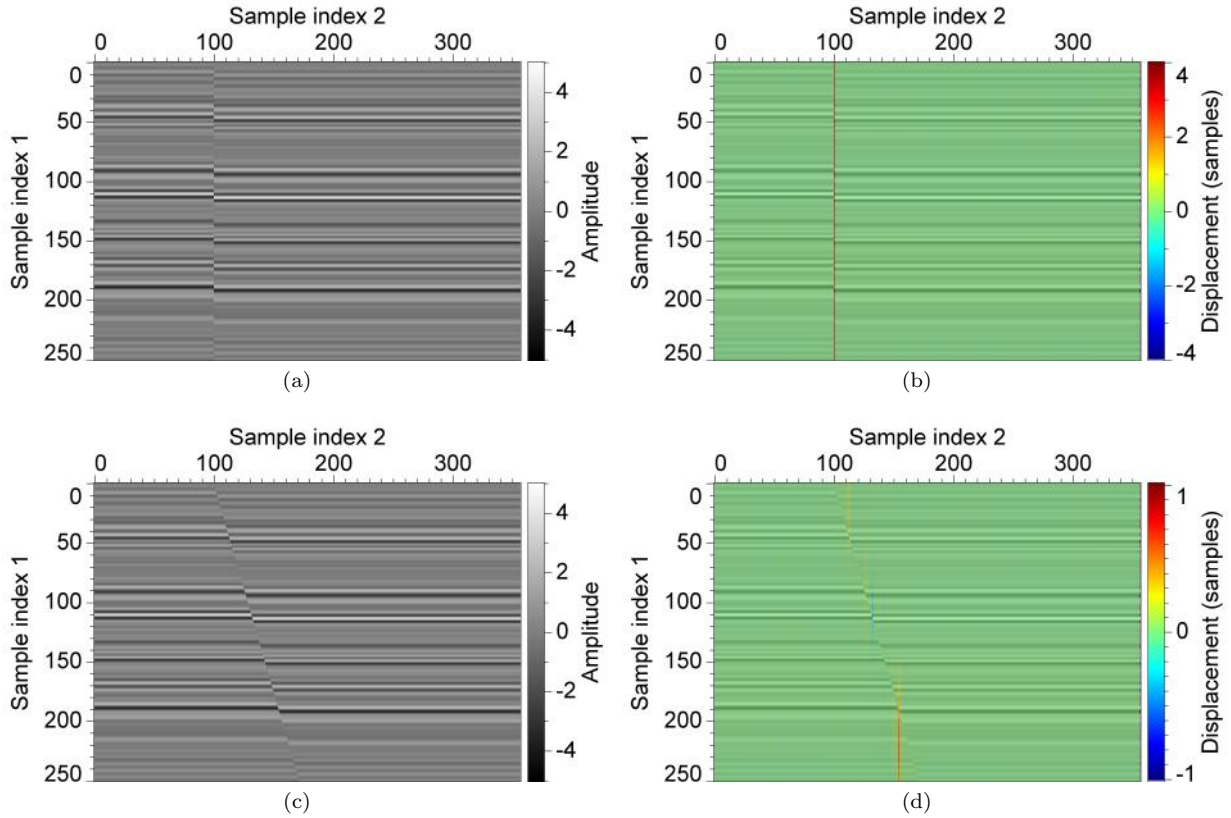


Figure 7. Crosscorrelation of adjacent pairs of traces in two seismic images with vertical (a) and non-vertical (c) faults yields corresponding estimates of the vertical components of displacements (b) and (d).

2.2 Faults are not vertical

Another problem with the simple trace-to-trace correlation method is that it performs poorly for faults that are not vertical. Each trace in a seismic image is a vertical sequence of pixels. To estimate displacements for non-vertical faults, we should crosscorrelate sequences of pixels that are parallel to the fault.

Figure 7 shows for a synthetic example the use of crosscorrelation of traces to estimate displacements. In this example, displacement is constant along the faults. As expected, displacements estimated from an image with a vertical fault are accurate. However, displacements estimated from image with the a non-vertical fault are inaccurate and inconsistent.

Figure 8 illustrates the source of this problem. Crosscorrelation of adjacent traces, vertical sequences

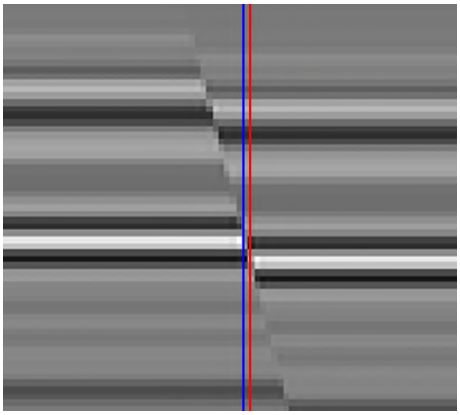


Figure 8. For a non-vertical fault, most samples in two adjacent traces (marked with red and blue lines) are similar, so that local trace-to-trace crosscorrelation fails to yield accurate estimates of even the vertical component of displacement.

of pixels, is inadequate, because so many of those pixels are identical for a fault that is not vertical.

2.3 Dipping layers are not faults

Faults need not be vertical and geologic layers also need not be horizontal. Displacements are nonzero for dipping structures. In Figure 5, vertical displacements vary slowly along both dipping layers and faulted areas. Which displacements correspond to faults?

Our goal is to determine the fault displacement vector field, which should be zero where faults do not exist. Locating faults in Figure 5 is even harder than locating faults in the original seismic image shown in Figure 1a.

Aimed to the three inadequacies mentioned above, we propose three improvements in following three sections in addition to the simple estimation.

3 FILTERING

Many papers describe filtering techniques for seismic images to aid interpretation (Luo et al., 2002; Fehmers and Höcker, 2003; Lu, 2006; AlBinHassan et al., 2006; Lu and Lu, 2009; Hale, 2009b). Most of these techniques enhance features, like structural layers, while suppressing noise. The Van Gogh filter (Fehmers and Höcker, 2003) is an application of a coherence-enhancing diffusion (Weickert, 1999) in geophysics. The key to this method is to solve a partial differential equation guided by a diffusion tensor field. Another seismic image filter is the edge-preserving filter, which was first proposed by Luo (2002), and then extended to 3D (AlBinHassan et al., 2006) by himself and his colleagues. The most significant advantages of the edge-preserving filter are that it is efficient and it is easy to implement.

However, none of these filters assist the correlations we require. We design a filter to assist the crosscorrelations between two sides of a fault. This filter gathers information from nearby traces, but does not gather information across the fault. Our filter generates two traces on each side of a fault by weighted averaging traces. The filter is designed as follows.

Left-to-right and right-to-left smoothing
Inputs:
seismic image $q(x_1, x_2)$ with $n_1 \times n_2$ samples
filter coefficient α
Outputs:
seismic image $q^-(x_1, x_2)$, filtered from left to right
seismic image $q^+(x_1, x_2)$, filtered from right to left
for all x_1
$q^-(x_1, 0) = q(x_1, 0)$
for $x_2 = 1, 2, ..., n_2 - 1$
for all x_1
find shifts $u(x_1)$ and
peak correlation coefficients $c_{max}(x_1)$,
for all x_1
use sinc interpolation to compute a shifted trace
$\tilde{q}(x_1) = q^-(x_1 + u(x_1), x_2 - 1)$
for all x_1
$a = \alpha \times c_{max}(x_1)$
$q^-(x_1, x_2) = a \times \tilde{q}(x_1) + (1 - a) \times q(x_1, x_2)$
By simply reversing the filtering direction, we get the right-to-left smoothing $q^+(x_1, x_2)$.

The **left-to-right and right-to-left smoothing** filter is an adaptive one-sided exponential filter (Oppenheim et al., 1999). The input of the filter is a 2D image, which is regarded as a set of 1D vertical traces. The image is processed trace-by-trace from left to right and right to left. The change made to the traditional one-sided exponential filter is to preserve the fault as much as possible during the smoothing. Normalized local crosscorrelation is used to find vertical shifts between consecutive traces. Traces are then warped with these shifts before they are used in the smoothing process. This operation is the key to avoid gathering information across faults before locating them. Coefficient α in this filter controls the effective length of the one-sided exponential filter (Oppenheim et al., 1999).

This filter is efficient because the one-sided exponential filter is the cheapest smoothing technique. In addition, the smoothing is confined by discontinuities between traces. By comparing the images in Figure 9 with the input image in Figure 1a, one sees that discontinuities are clearer and layers do not extend across the fault.

In practice, we compute displacement fields by picking one trace from the left-to-right smoothed image shown in Figure 9a and another trace from the right-to-left smoothed image shown in Figure 9b. This conforms to the process that a pair of eyes uses to find a fault: first scanning the image from left to right and then from

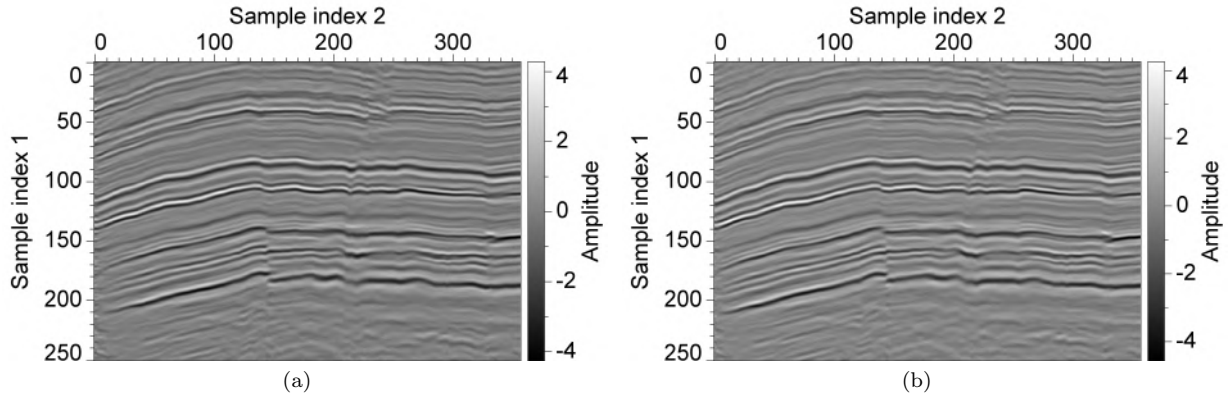


Figure 9. Filtering the image shown in Figure 1a from left to right (a) and from right to left (b).

right to left, and locating the fault curves as observed differences.

4 SHEARING

When faults are not vertical, we should correlate non-vertical sequences of pixels, as shown in Figure 10. Unfortunately, we do not yet know the locations and orientations of faults. One possible way to solve this problem is to search for all orientations on each pixel of the image. Therefore, a filter that hence rotates the window locally is a plausible way. However, the computation of rotation around one pixel does not contribute to computing rotations in other places. This fact makes the orientation search time-consuming.

Although the rotation-based search method is expensive, it gives us a clue to reduce the cost. Paeth (1990) implements the rotation of an image by consecutively shearing the image three times. The shear transformation is an affine transformation where one coordinate of each point is changed in proportion to its another coordinate. In 2D, we have two kinds of shear transform: horizontal shear (x coordinates change in proportion to y coordinates) and vertical shear (y coordinates change in proportion to x coordinates), see Figure 11.

As shown in Figure 12, we shear the seismic image horizontally to make the faults vertical. We define an integer l_1 as the maximum vertical displacement of a fault and a float s as the shear amount of the image that controls the number of samples will be moved. Mathematically, shearing a 2D scalar field $f(x_1, x_2)$ is to create a new 2D scalar field $f'(x_1, x_2)$ and

$$f'(x_1, x_2) = f(x_1, x_2 - sx_1), \tag{6}$$

where $s > 0$ indicates the rows will be moved to the right, $s < 0$ indicates the rows will be moved to the left. $s \in (-1, 1)$ because we stipulate that a fault in our model must form less than a 45 degree angle from the vertical line. To sample the range of shear amount

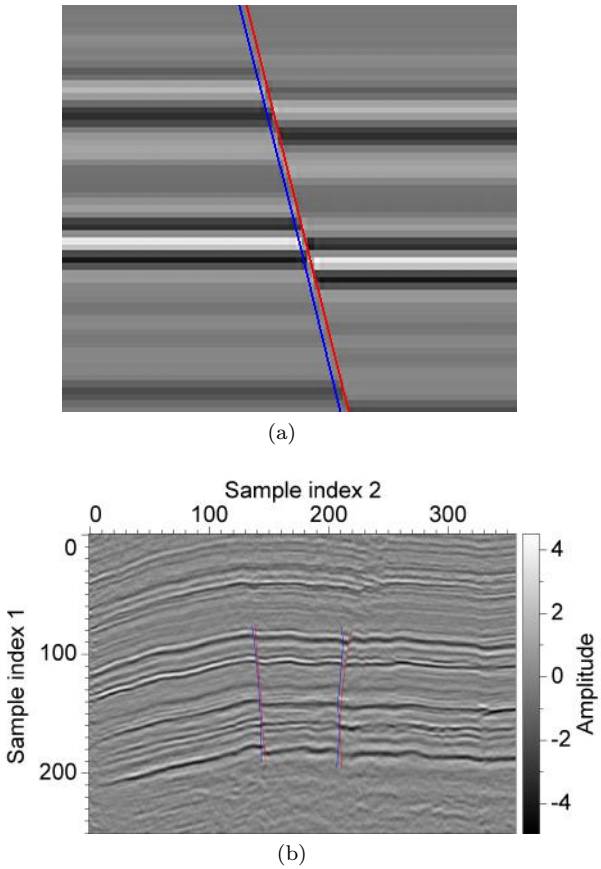
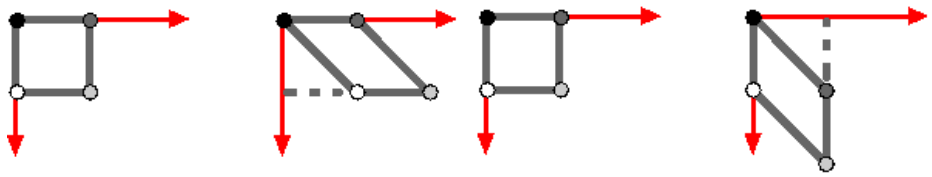


Figure 10. Ideal trace-picking in a non-vertical fault zone: (a) the magnification of a part of 7c; (b) the ideal trace picking for the image in Figure 1a.

$(-1, 1)$, we define another integer $l_2 \in (-l_1, l_1)$ and $s = \frac{l_2}{l_1}$. Since s is not an integer, we should translate the sequence (row) by a float amount. We use sinc interpolation to perform this sub-pixel translation.

The shear transform may move out and truncate



(a) (b)

Figure 11. Shear transform: (a)horizontal shear; (b) vertical shear.

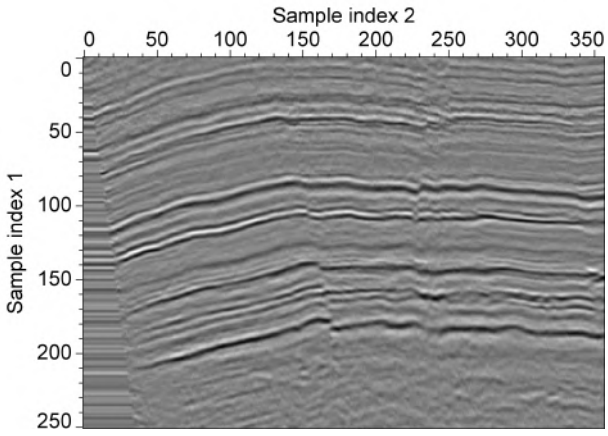


Figure 12. By shearing the image shown in Figure 1a with shear amount $s = \frac{2}{15}$, we make faults at some places vertical.

a part of the input image. To avoid losing data after shearing, we extrapolate the image before shearing it.

5 LOCATING

We assess fault locations through a two-stage pattern analysis. In the first stage of analysis, we estimate the location of a rough fault zone; in the second stage, we pick out the precise one-pixel-wide fault out from the fault zone.

In a seismic image which contains only horizontal geologic layers, vertical displacements between traces are zeros before and after shearing. However, the shearing may change the displacement between traces when the layers are dipping. As shown in Figure 13, f and f' are images before and after shearing.

We model a dipping geologic layer at a certain point as a line segment. Therefore, f becomes a univariate function g:

$$f(x_1, x_2) = g(x_1 - px_2), \tag{7}$$

where p is the slope. As shown in equation 6, $f'(x_1, x_2) = f(x_1, x_2 - sx_1)$, therefore

$$\begin{aligned} f'(x_1, x_2) &= g[x_1 - p(x_2 - sx_1)] \\ &= g[(1 + sp)x_1 - px_2]. \end{aligned} \tag{8}$$

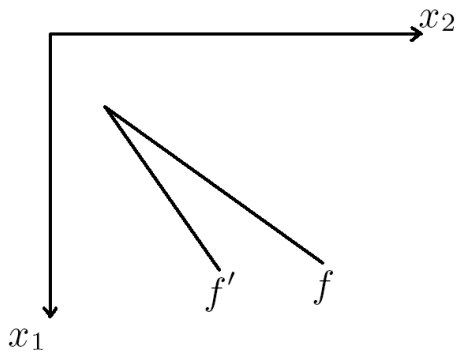


Figure 13. Shearing a dipping layer.

Since f represents a line segment, we have

$$\triangle f = \frac{\partial f}{\partial x_1}\triangle x_1 + \frac{\partial f}{\partial x_2}\triangle x_2 = 0. \tag{9}$$

Thus,

$$p = \frac{\triangle x_1}{\triangle x_2} = -\frac{\frac{\partial f}{\partial x_2}}{\frac{\partial f}{\partial x_1}}. \tag{10}$$

Similarly, we have

$$p' = -\frac{\frac{\partial f'}{\partial x_2}}{\frac{\partial f'}{\partial x_1}}, \tag{11}$$

where p' is the slope of the layer after shearing. Substituting f' in equation 11 by equation 8, we obtain

$$\begin{aligned} p' &= -\frac{\frac{\partial g[(1+sp)x_1 - px_2]}{\partial x_2}}{\frac{\partial g[(1+sp)x_1 - px_2]}{\partial x_1}} \\ &= -\frac{-p\frac{\partial g}{\partial x_2}}{(1+sp)\frac{\partial g}{\partial x_1}} \\ &= \frac{p}{1+sp}. \end{aligned} \tag{12}$$

Here, p' is a function of the shear amount s. Note that slopes p and p' are also vertical displacements right at the selected point before and after shearing, respectively. When $p = 0$, $p'(s) = p = 0$, which indicate that the shearing does not affect the horizontal layer.

The above analysis reveals the effect of shearing on a dipping layer where no fault exists. However, in a

zone where a fault exists, the function $p'(s)$ does not follow the function shown in equation 12. Because $s = \frac{l_2}{l_1}$ and l_1 is a constant, $p'(s)$ is sampled according to l_2. Figure 14 illustrates different patterns of function relationships. One can see a peak in the black curve in Figure 14b, which corresponds to the black pixel in Figure 14a. This pixel is right on a fault. In Figure 14b, the horizontal coordinate l_2 of the peak indicates that shearing the image by amount $\frac{l_2}{l_1}$ makes the fault vertical. Curves in Figure 14c approximately conform to the relationship shown in equation 12. Figure 15 illustrates such relationship in real data. We can still distinguish the blue curve from others.

In the first stage, we distinguish samples around a fault with samples far away from a fault according to the following criterion: if shifts vary slowly according to different shearings, there is no fault. In other words, if there is an outstanding point in the displacement-shear curve, as the black curve shown in Figure 14, there is a fault.

In practice, we set a fault threshold η and compute the average value $ave(p'(s))$ as well as the maximum value $\max(p'(s))$ of vertical displacements with different shearings at a particular point. If $\max(p'(s)) > \eta$ and $\max(p'(s)) > c|ave(p'(s))|$, there exists a fault. The coefficient c must be chosen carefully to distinguish the curves which are not slowly varying. We name c as the fault coefficient. The first-stage pattern analysis yields fault displacement zones shown in Figure 16a.

A second-stage pattern analysis is required to pick one-pixel-wide fault curves from the fault zones. We plot the displacement-to-shearing function for five consecutive points along a horizontal line around fault location $(105, 140)$. Five curves share a same pattern. However, only one of these five samples has fault. We choose the yellow point because its corresponding curve has the largest maximum displacement value. By scanning horizontally across every fault zones, we get the fault displacement field as shown in Figure 16b.

5.1 Compute displacement vector

After we obtain the vertical shift v_1 and the shear amount s, we can compute the displacement vector

$$u_1(x_1, x_2) = v_1(x_1, x_2 - sx_1),$$
$$u_2(x_1, x_2) = -su_1(x_1, x_2).$$

The horizontal components of the fault displacement vector field is shown in Figure 18a. Signs of horizontal displacements indicate that geologic layers have been displaced either rightward (positive) or leftward (negative). By combining the horizontal components with the vertical components (shown in Figure 18b), we finally get the fault displacement vector field.

6 PARAMETERS

Because several changeable parameters exist in our algorithm (c, η and l_1), we illustrate the effect of changing these coefficients in this section. The standard choices of these coefficients are $c = 0.25$, $\eta = 0.4$ and $l_1 = 15$ for the image shown in Figure 1a. When illustrating the effect of changing one of these coefficients, we keep two others unchanged.

6.1 The fault coefficient c

The fault coefficient c largely controls the identification of a fault. From Figure 14, one can easily tell the difference between the pattern of fault place and the pattern of non-fault place. Consequently, c can be chosen in a large range in this synthetic image. However, in the real data, as shown in Figure 15, the pattern corresponding to the fault place is less distinguish. In this case, the possible range of c is largely confined. One can see some incorrect identifications of faults in the left half of the image in Figure 19a due to a relatively small value for c. When c increases, these inaccurate picks disappear, as shown in Figure 19b and c. However, the fault located at $(105, 140)$ is not identified in Figure 19b and c.

Because (1) seismic images are often contaminated by random noise and (2) dips are varying along geologic layers, an adaptive c or a more sophisticated pattern analysis is required for processing seismic images with lower illumination qualities.

6.2 The fault threshold η

Choosing different fault thresholds η shows us different levels of detail for fault displacement fields. (Note Figure 20a, b, c and d.) A reasonable fault threshold is related to the sampling interval of the image. Interpreters can change the fault threshold interactively to get the most satisfactory result.

6.3 The maximum vertical displacement l_1

The continuity of fault curves is related to the maximum vertical displacement l_1. As mentioned above, we must investigate the problem in a relatively large area. l_1 controls the vertical window size, thus cannot be set too small. Figure 21a, b and c illustrate this effect on the displacement field by changing l_1.

As Figure 21 demonstrates, a larger l_1 value yields more continuous fault curves. However, shorter faults with rapidly changing orientation can be well detected using a smaller l_1. The reason is that using a larger l_1 tends to catch longer faults. Detailed differences between $l_1 = 12$ and $l_1 = 18$ can be found in Figure 22.

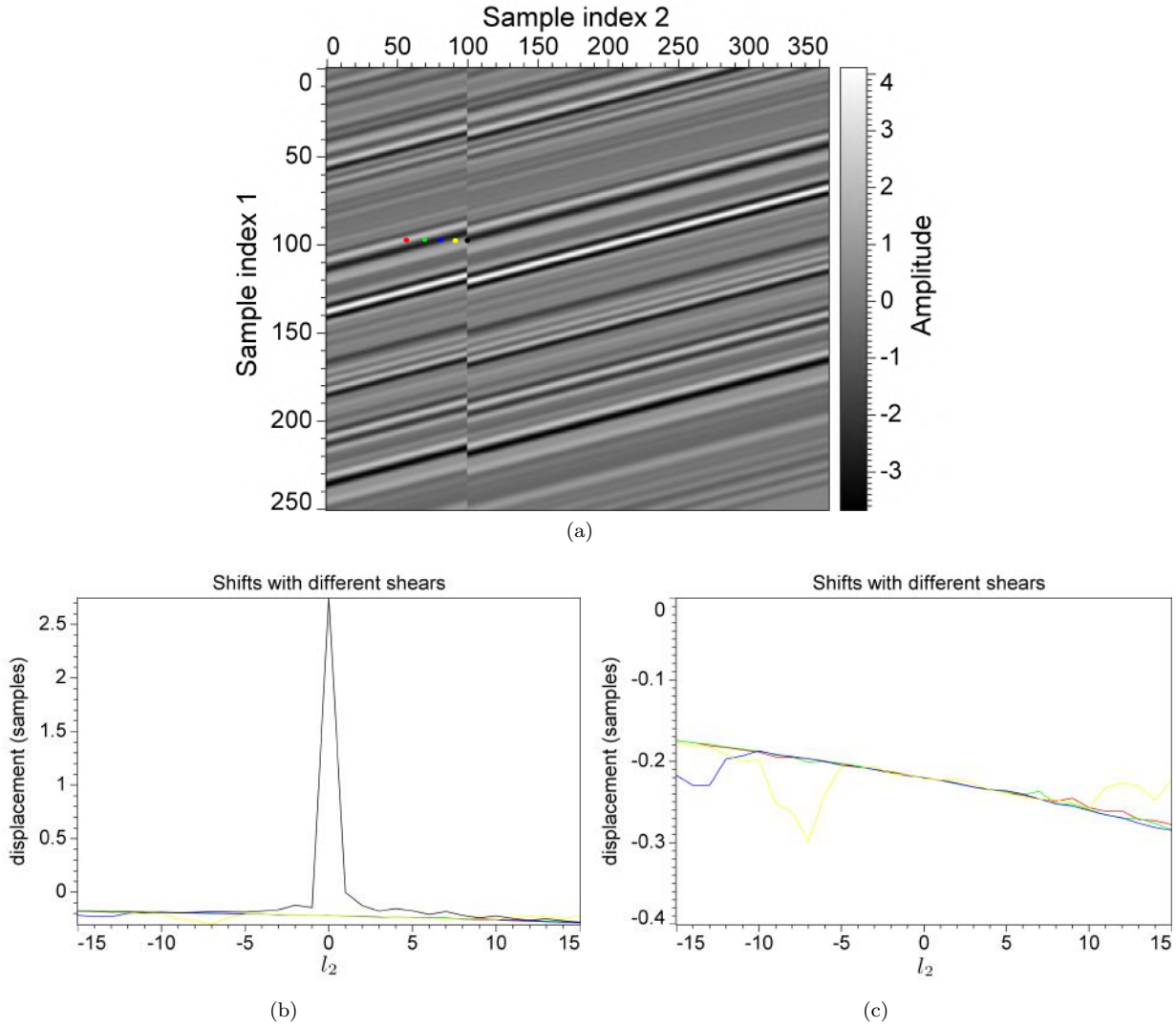


Figure 14. Investigating the function relationship between the vertical displacement and l_2. Five curves in (b) correspond to five points selected from the synthetic image (a) with the same colors. Leaving the black curve (b) out, we get similar patterns (c).

7 CONCLUSION

Our estimation of fault displacement vector fields in seismic images consists of four primary steps:

- filter the seismic image;
- shear the image to make faults vertical;
- estimate fault displacements from the sheared images using normalized local crosscorrelation;
- apply two-stage pattern analysis to exactly locate faults.

By following these steps sequentially, one obtains fault curves as well as the displacement vector field defined in section 1.1. If one omits the second-stage pattern analysis, one can get a fault displacement zone which is similar to the fault zone model estimated by Kadlec et al. (2008).

The fault displacement vector field estimated by our method has a sub-pixel precision in the vertical and horizontal components but does not have sub-pixel precision fault locations. Exact faults are usually located between two pixels.

This fault displacement vector field can be used to improve estimating the structure tensors (Hale, 2009b) at places around a fault. These tensors can be further used to adaptively smooth the seismic image or guide interpolation. Another potential usage is to place streamlines on seismic images, which is similar to the auto-

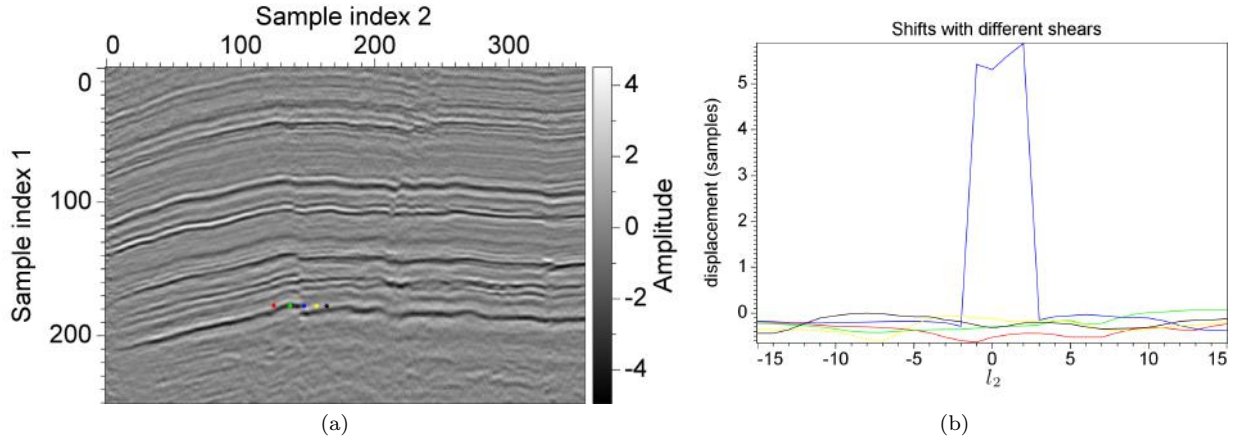


Figure 15. Investigating the function relationship between the vertical displacement and l_2. Five curves (b) correspond to five points selected from the synthetic image (a) with the same colors.

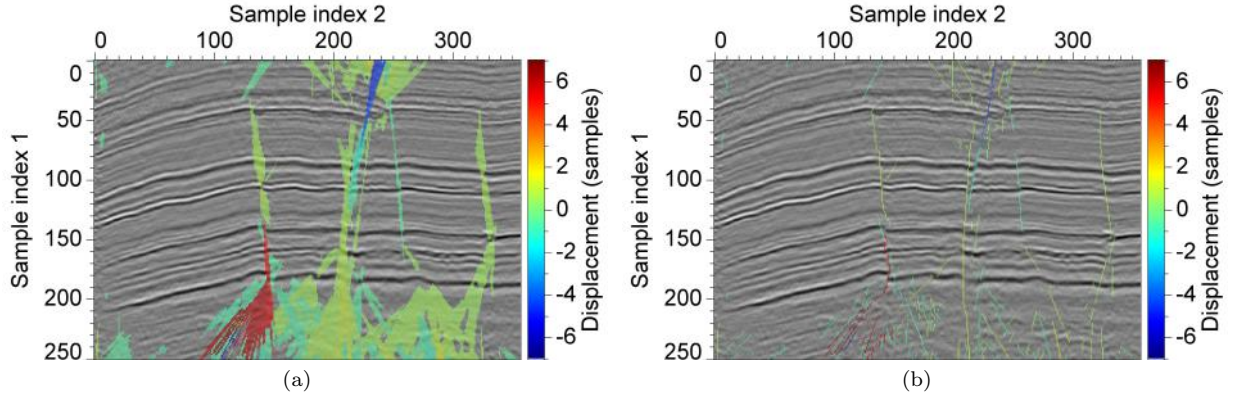


Figure 16. Applying the second stage pattern analysis on the fault displacement zone (a), we obtain the fault displacement field (b).

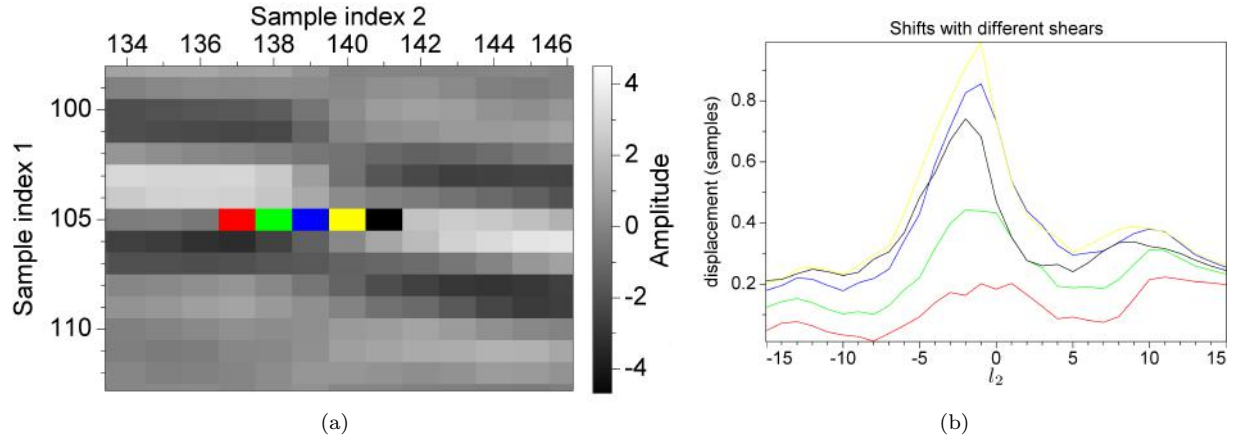


Figure 17. Investigating the function relationship between the vertical displacement and l_2 in the fault zone around $(105, 140)$. Five curves (b) correspond to five points selected from the image (a) with same colors.

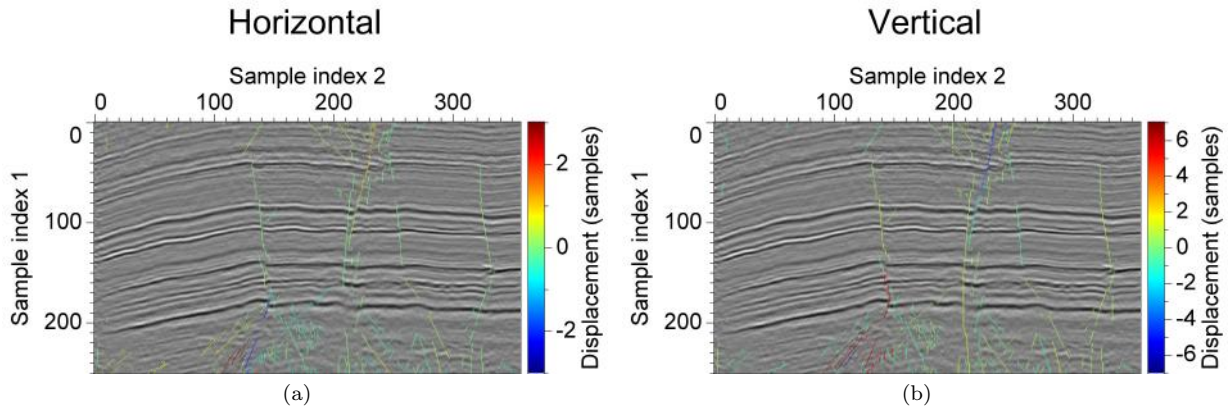


Figure 18. Fault displacement vector field: (a) horizontal components; (b) vertical components;

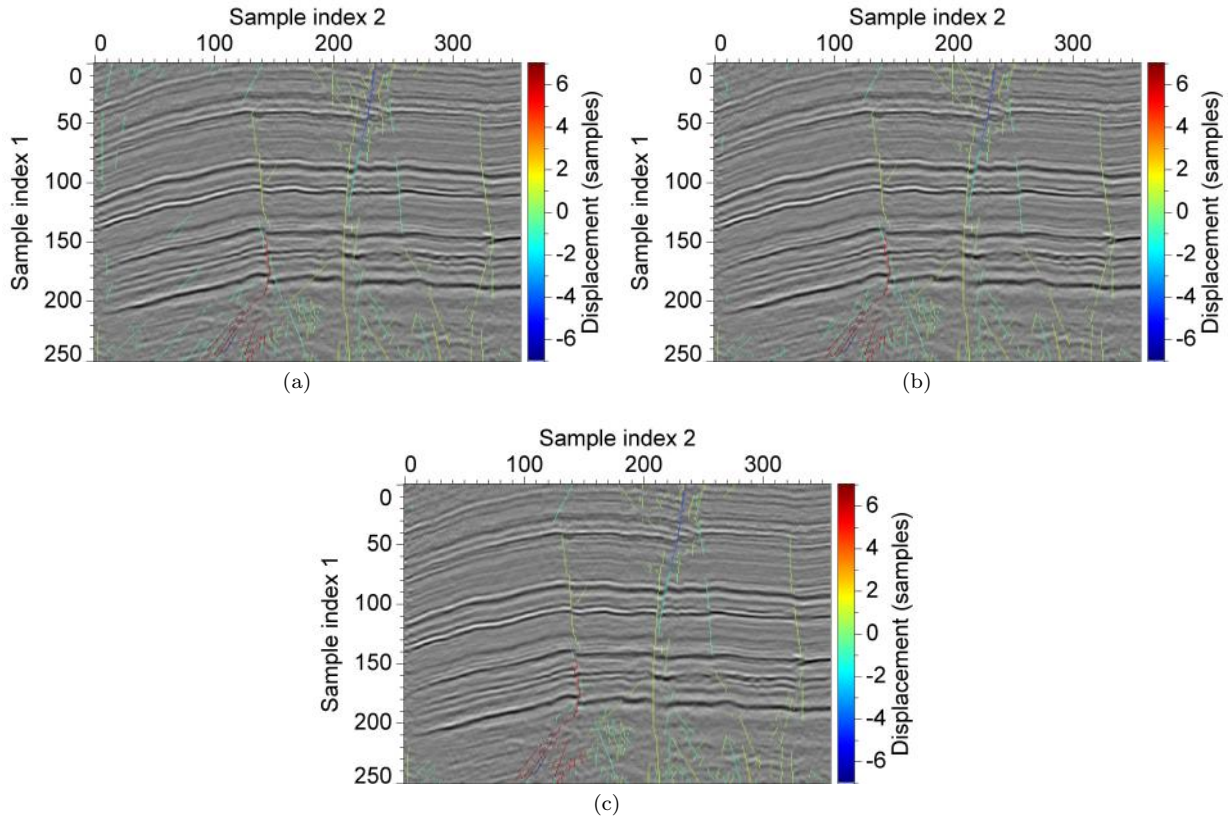


Figure 19. Vertical components of displacement vector fields estimated by using different fault coefficient c: (a) 0.2; (b) 0.25; (c) 0.3.

matic horizon picking. Furthermore, the fault displacement vectors can also used in flattening seismic images.

ACKNOWLEDGMENTS

Thanks to the Rocky Mountain Oilfield Test Center, a facility of the U.S. Department of Energy, for providing the seismic image used here. We would like to thank our colleagues for their discussions and feedback on this

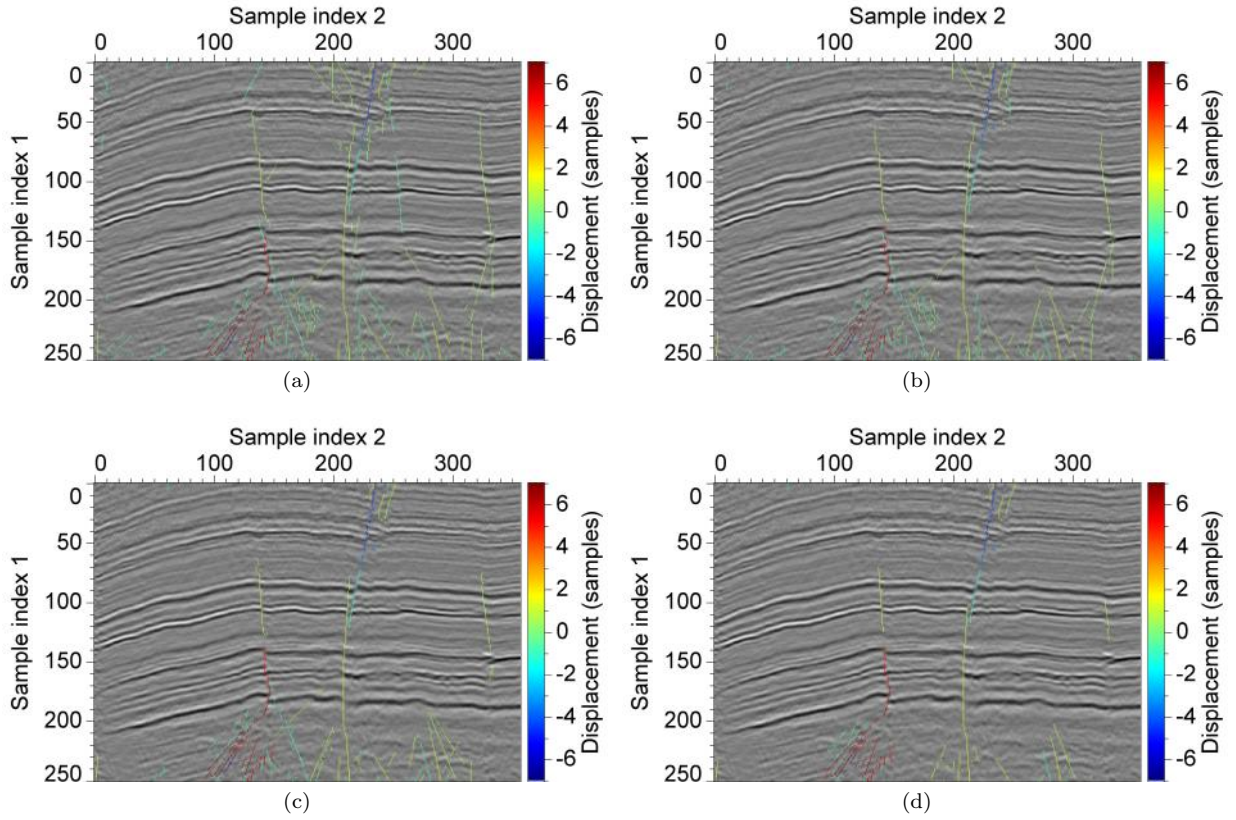


Figure 20. Vertical components of displacement vector fields estimated by using different fault threshold η: (a) 0.4; (b) 0.5; (c) 0.6; (d) 0.7.

research. We especially thank Diane Witters for her help with writing.

REFERENCES

Admasu, F., and K. Tönnies, 2006. Multi-scale bayesian based horizon matchings across faults in 3d seismic data *Proc. Deutsche Arbeitsgemeinschaft für Mustererkennung DAGM2006, Lecture Notes in Computer Science*, **4174**, 384–393.

AlBinHassan, N. M., Y., Luo, and M. N. Al-Faraj, 2006. 3D edge-preserving smoothing and applications. *Geophysics*, **71**(4), 5–11.

Al-Dossary, S., and K. J. Marfurt, 2006. 3D volumetric multispectral estimates of reflector curvature and rotation. *Geophysics*, **71**(5), 41–51.

Aurnhammer, M., and K. Tönnies, 2005. A genetic algorithm for automated horizon correlation across faults in seismic images. *IEEE Transactions on evolutionary computation*, **8**(5), 201–210.

Bahorich, M., and S. Farmer, 1995. 3-D seismic discontinuity for faults and stratigraphic features: The coherence cube. *The Leading Edge*, **14**(10), 1053–1058.

Barnett, J. A. M., J. Mortimer, J. H. Rippon, J.J. Walsh and J. Watterson, 1987. Displacement geometry in the volume containing a single normal fault. *The American Association of Petroleum Geologists Bulletin*, **71**(8), 925–937.

Cohen, I. and R. R. Coifman 2002. Local discontinuity measures for 3-D seismic data. *Geophysics*, **67**(6), 1933–1945.

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**(9), 2094–2102.

Faraklioti, M., and M. Petrou 2004. Horizon picking in 3D seismic data volumes. *Machine Vision and Applications*, **15**, 216–219.

Fehmers, G. C. and C. F. Höcker, 2003. Fast structural interpretation with structure-oriented filtering. *Geophysics*, **68**(4), 1286–1293.

Hale, D., 2006, Fast local cross-correlations of images. *SEG Expanded Abstracts*, **25**, 3160–3164

Hale, D., 2009a. A method for estimating apparent displacement vectors from time-lapse seismic images. *Geophysics*, **74**(5), 99–107.

Hale, D., 2009b, Structure-oriented smoothing and semblance: CWP Report 635, 261–270

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 Geoscience*, **13**, 133–138.

Liu, G., S. Fomel, L. Jin, and X. Chen, 2009, Stacking seismic data using local correlation, *Geophysics*, **43**(3), 43–48.

Lu, W., 2006, Adaptive noise attenuation of seimic images

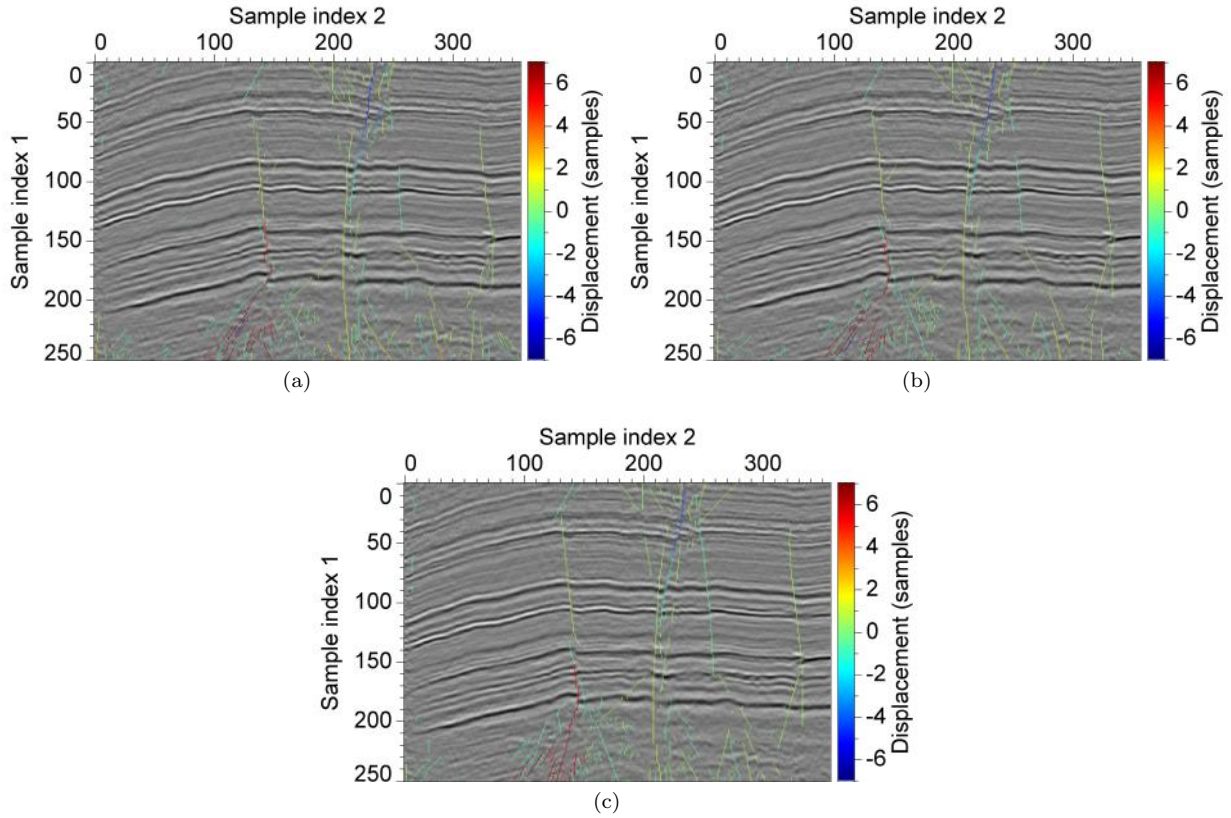


Figure 21. Vertical components of displacement vector fields estimated by using different maximum vertical displacement l_1: (a) 12; (b) 15; (c) 18.

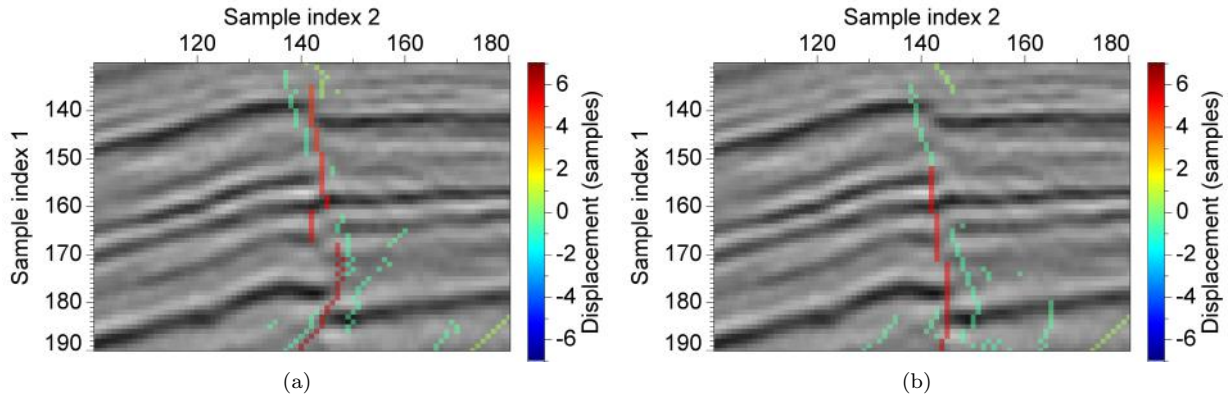


Figure 22. Subsets of vertical displacement fields estimated by using different maximum vertical displacement l_1: (a) 12; (b) 18.

based on singular value decomposition and texture direction detection, *Journal of Geophysics and Engineering*, **3**, 28–34.

Lu, Y. and W. Lu, 2009, Edge-preserving polynomial fitting method to suppress random seismic noise, *Geophysics*, **74**(4), 69–73.

Luo, Y., M. Marhoon, S. Al-Dossary, and M. Alfaraj, 2002, Acquistion ProcessingEdge-preserving smoothing and applications, *The Leading Edge*, **21**(136), 136–141.

Oppenheim, A. V., R. W. Schafer, and J. R. Buck, 1999. Discrete-Time Signal Processing (2nd Edition). *Prentice Hall*

Patel, D., C. Giertsen, J. Thurmond, and M.E. Gröller, 2009. The seismic analyzer: Interpreting and illustrating 2D

Seismic Data. *IEEE Transactions on Visualization and Computer Graphics*, **14**(6), 1571–1578.

Patel, D., Ø. Sture, H. Hauser, C. Giertsen and M.E. Gröller, 2009. Knowledge-assisted visualization of seismic data. *Computer & Graphics*, **33**, 585–596.

Parth, A. W., 1990. A fast algorithm for general raster rotation. *Graphics Gems*, 179–195, Academic Press.

Tüceryan, M. and A. K. Jain, 1990. Texture segmentation using voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 211–216.

Weickert, J., 1999. Coherence-Enhancing Diffusion Filtering. *International Journal of Computer Vision*, **31**(2/3), 111–127.

Watterson, J., 1986. Fault dimensions, displacements and growth. *Pure and Applied Geophysics*, **124**, 365–373.

Fault_(geology). http://en.wikipedia.org/wiki/Fault_(geology).