Fault surfaces and fault throws from 3D seismic images

Dave Hale, Center for Wave Phenomena, Colorado School of Mines

SUMMARY

A new method for processing 3D seismic images yields images of fault likelihoods and corresponding fault strikes and dips. A second process automatically extracts from those images fault surfaces represented by meshes of quadrilaterals. A third process uses differences between seismic image sample values alongside those fault surfaces to automatically estimate fault throw vectors. While some of the faults found in one 3D seismic image have an unusual conical shape, displays of unfaulted images illustrate the fidelity of the estimated fault surfaces and fault throw vectors.

INTRODUCTION

Fault surfaces like those shown in the close-up views of Figure 1 are an important aspect of subsurface geology that can be derived from seismic images. Therefore, various fault tracking methods, including those proposed by Pedersen et al. (2002, 2003), Admasu et al. (2006), Kadlec et al. (2008) and Kadlec (2011), have been developed to extract such surfaces.

The fault throws shown in Figure 1 are important as well, as they enable correlation of subsurface properties across faults. Among methods developed to estimate fault throws are those described by Borgos et al. (2003), Aurnhammer and Tönnies (2005) and Admasu (2008).

Figure 2 provides more extensive views of many fault surfaces and corresponding fault throws computed for the same 3D seismic image. Also shown in Figure 2 are images after *unfaulting*, using a process described by Luo and Hale (2012). After unfaulting, seismic reflections are generally more continuous across faults, suggesting that estimated fault throws are consistent with true fault displacements.

This paper describes a sequence of three new methods to (1) compute 3D fault images, (2) extract fault surfaces, and (3) estimate fault throws. I used this three-step sequence to compute the fault surfaces and throws displayed in Figures 1 and 2. Although each of the three steps was designed in conjunction with the others in this sequence, aspects of any one of them could be used to enhance other methods cited above.

FAULT IMAGES

Before extracting fault surfaces like those shown in Figures 1 and 2, I first compute images of faults. The method I use for this first step is based on semblance (Taner and Koehler, 1969), and is therefore similar to methods proposed by Marfurt et al. (1998). Like Marfurt et al. (1999), I compute semblances from small numbers (3 in 2D, 9 in 3D) of adjacent seismic traces, after aligning those traces so that any coherent events are horizontal.

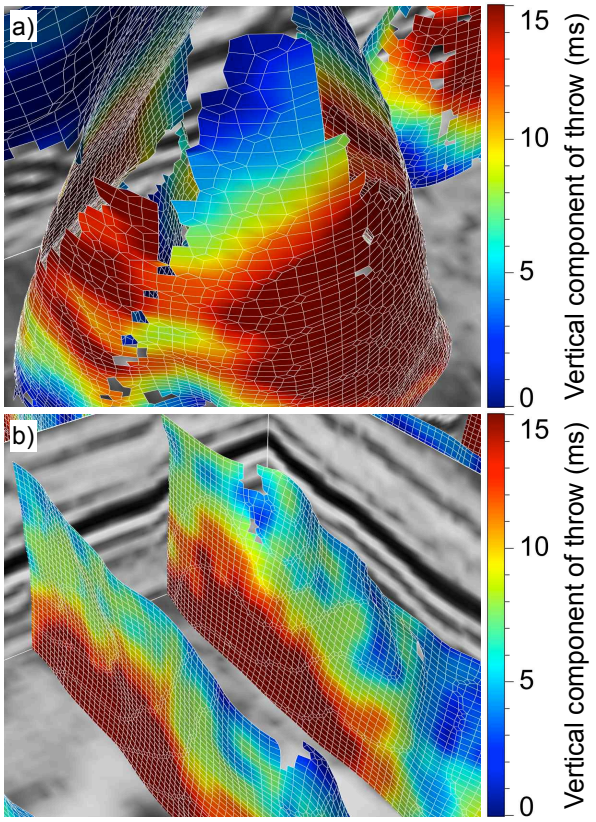

Figure 1: Close-up views of roughly conical (a) and planar (b) fault surfaces and fault throws computed automatically from a 3D seismic image. Vertical and horizontal image slices are shown in the background. Vertical fault throws are measured in ms because the vertical axis of the image is time. Each quadrilateral intersects exactly one edge in the 4 ms by 25 m by 25 m image-sampling grid.

Semblance is a measure of coherence in the range $[0,1]$; it is a normalized ratio, the square of an average value divided by an average of squared values. Because faults are most likely to exist where semblance s is low, I (somewhat arbitrarily) define and compute a measure of fault likelihood $f = 1 - s^8$.

When used to highlight faults, some sort of averaging or smoothing of semblance (or some other attribute) is required, as emphasized by Gersztenkorn and Marfurt (1999) and Aqrawi and Boe (2011). These authors describe vertical smoothing of fault attributes.

However, faults need not be vertical. Therefore, when averaging the numerators and denominators of normalized semblance ratios, I vary the orientation of the smoothing filter in a scan over possible fault orientations.

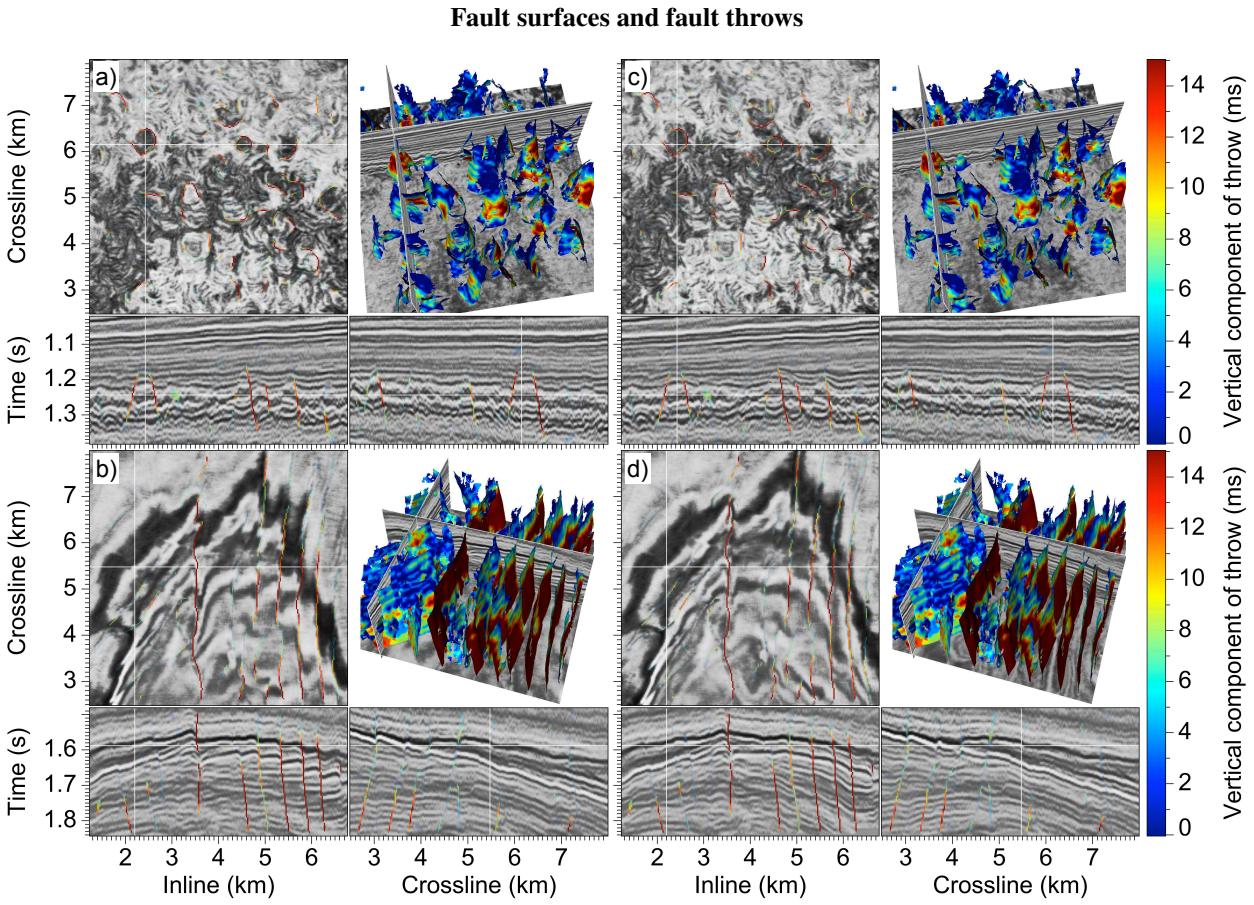

Figure 2: Fault surfaces and fault throws computed for two different subsets of a 3D seismic image. Faults extracted from the shallower subset (a) have conical shapes, while those extracted from the deeper subset (b) have more typical planar shapes. Seismic reflectors are more continuous in the corresponding unfaulted images (c and d).

Figure 3 illustrates for a 2D image the results of non-vertical smoothing for two different fault dips θ in this scan. This example shows that fault likelihoods tend to be largest when the smoothing of semblance numerators and denominators is aligned with the faults, which are not vertical.

Much like Cohen et al. (2006), I scan over multiple fault strikes and dips to determine the orientation that maximizes fault likelihood. In the 3D examples shown in this paper, this scan included $N_\phi = 26$ fault strikes ϕ and $N_\theta = 22$ fault dips θ, for a total of $N_\phi N_\theta = 572$ possible fault orientations.

The computational cost of the scan is proportional to this number of orientations. To reduce this cost, for each orientation I use efficient recursive smoothing filters to perform the averaging of semblance numerators and denominators. The computational cost of these recursive filters is independent of the spatial extent of their impulse responses, which may include well over 1000 samples in 3D images. This number represents the number of samples that contribute to the computation of fault likelihood for one orientation at one sample location in a 3D image (Cohen et al., 2006). With recursive smoothing filters I avoid this large factor in computational cost.

Figure 4a shows fault likelihoods computed with a scan over $N_\theta = 22$ fault dips for the 2D seismic image in Figure 3a. Ridges of fault likelihood in this *fault image* generally coincide with faults apparent in the seismic image. These ridges can be found by simply scanning each row of the fault image, preserving only local maxima, and setting fault likelihoods elsewhere to zero, as shown in Figure 4b. In effect, this process thins the fault image, significantly reducing the number of image samples at which a fault may be considered to exist.

It is important to remember that for any images of fault likelihood, such as those shown in Figure 4, we have corresponding images of fault orientation, the fault strikes and dips for which fault likelihood is maximized. These images of fault orientation are especially useful when extracting fault surfaces from 3D images.

FAULT SURFACES

For 3D seismic images, ridges of fault likelihood correspond to potential fault locations. However, it is more difficult to extract ridge surfaces from 3D images than to extract ridge curves from 2D images as illustrated in Figure 4.

I extract fault surfaces from 3D images of fault likelihoods f,

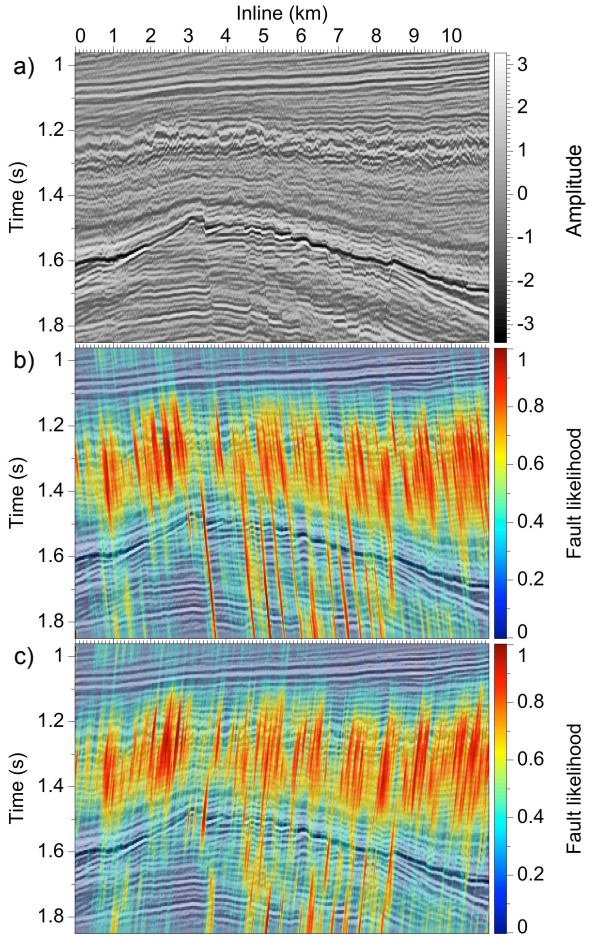

Figure 3: Fault likelihoods computed for a 2D seismic image (a) and for two different fault dips θ, one positive (b) and the other negative (c), in the scan used to estimate fault likelihoods and orientations.

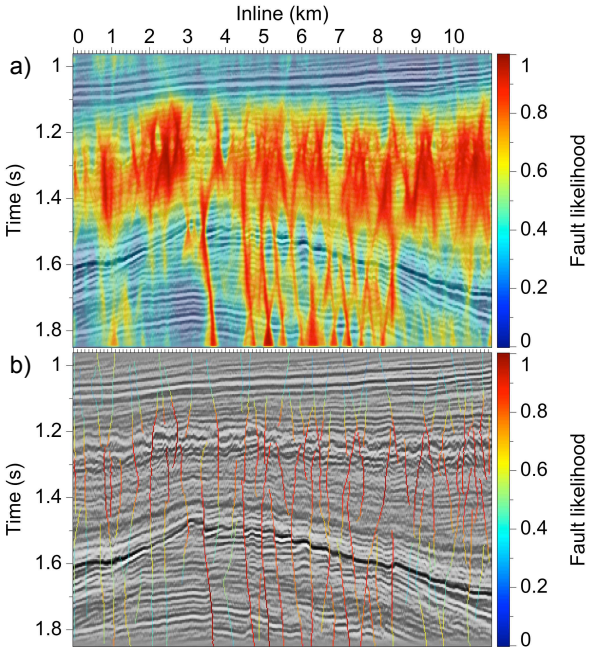

Figure 4: Fault likelihoods computed by scanning over fault dips θ, before (a) and after (b) ridge extraction.

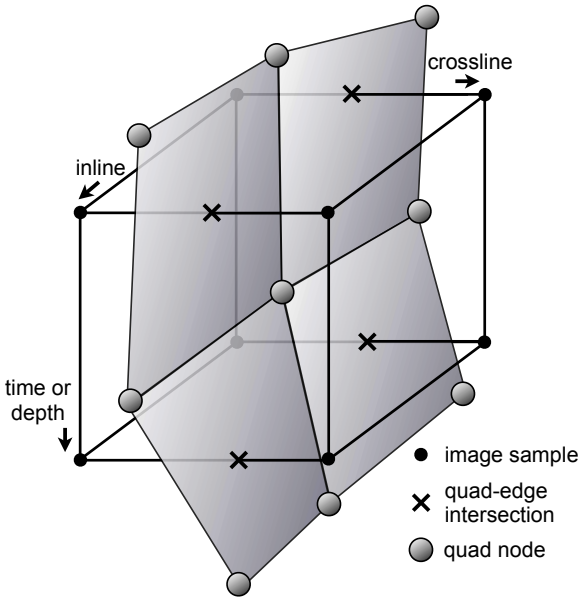

Figure 5: Four adjacent quads in a fault surface share a node that lies within one cell of the 3D fault image sampling grid. Spatial coordinates of the quad node are averages of the coordinates of intersections of the fault surface and edges of the image sampling grid.

strikes ϕ, and dips θ using a method for extracting ridge surfaces similar to that proposed by Schultz et al. (2010), which they demonstrate for 3D medical images of the human brain. The fault surfaces shown in Figures 1 and 2 are ridges in 3D images of fault likelihood, and are represented by meshes of quadrilaterals (hereafter referred to as quads).

As shown in Figure 5, each quad in a fault surface intersects exactly one edge of the 3D sampling grid for the fault image. Each of the four nodes of a quad lies within exactly one cell of that grid. The coordinates of a quad node within any such cell are averages of the coordinates of all quad-edge intersections for that cell. This averaging enables representation of a fault surface with sub-voxel precision. Therefore, to find the locations of the quad nodes, we must first find the intersections of the fault surface and edges of the 3D sampling grid.

I find those surface-edge intersections using an adaptation of the method proposed by Schultz et al. (2010), in which I ensure that the orientations of ridge surfaces extracted from 3D images of fault likelihoods are consistent with the corresponding 3D images of fault strikes and dips computed during the scan described above.

I have made no attempt to fill any of the small holes apparent

in these surfaces, although such a filling process would be easy to implement because each quadrilateral is linked to its neighbors. The fact that holes are small is due to the continuity of ridges in the 3D images of fault likelihood.

FAULT THROWS

Because each quad in fault surfaces like those shown in Figure 1 corresponds to exactly one edge in the 3D image sampling grid, it is straightforward to walk up and down fault curves and gather samples of the 3D seismic image on opposite sides of a fault. I then compute fault throws that minimize sums of squared differences of those sample values.

This new method for computing fault throws is an adaptation of a classic dynamic programming solution (Sakoe and Chiba, 1978) to the problem of automatic speech recognition. That solution today is often called *dynamic time warping* and is here adapted to find a spatial warping that best aligns samples of 3D seismic images alongside faults, as illustrated in Figure 2.

One of the most attractive features of the dynamic time warping algorithm is that it optimally aligns two time series while constraining the amount of stretching or squeezing of sequences that is permitted during alignment. The relative shift (here, fault throw) between two sequences may vary with time (or depth), but dynamic time warping constrains the rate at which the shift changes with time.

My adaptation of this algorithm is to constrain the rate at which fault throw varies in both strike and dip directions along a fault. This constraint is much like that those imposed in dynamic image warping (e.g., Pishchulin, 2010), in which shifts are constrained to vary slowly in both horizontal and vertical directions.

However, we cannot simply estimate fault throws by aligning a 2D image extracted from the footwall side of a fault surface with another 2D image extracted from the hanging-wall side. Consider for example the fault surface shown in Figure 6, where part of the surface lies in front of another part of that same surface. This situation occurs often in the fault surfaces shown in Figure 2. Nevertheless, quad meshes provide the left-right and up-down connectivity required to constrain changes in fault throws in both the strike and dip directions within such surfaces.

CONCLUSION

The methods proposed in this paper were designed as parts of a three-step process to (1) compute images of fault likelihood, strike and dip, (2) extract fault surfaces, and (3) estimate fault throws.

It is significant that the scan in the first step yields images of fault strikes and dips for which fault likelihood is maximized. These estimates of fault orientations are useful in several consistency tests performed in the second step used to extract fault surfaces.

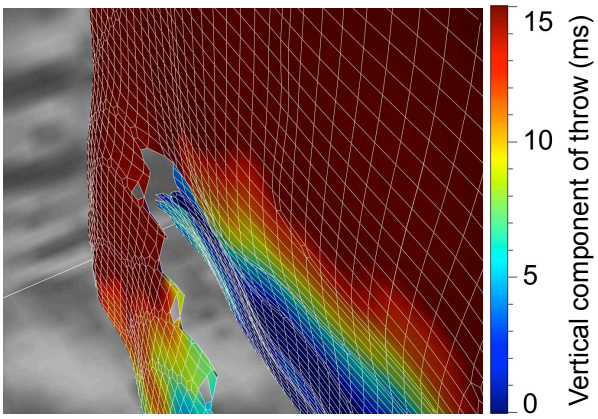

Figure 6: Close-up view of fault throws computed for a fault surface in which one part of the surface lies in front of another part. For such surfaces we cannot simply compute throws from footwall and hanging-wall images extracted alongside faults.

The quad-mesh representation for those fault surfaces facilitates the third step of estimating fault throws. Because throw vectors connect image samples on one side of a fault to those on the other side, it is especially convenient that a quad in the fault surface lies between two adjacent samples of the seismic image. In addition, the quad mesh provides left-right and up-down connectivity needed to implement the dynamic warping algorithm used to estimate fault throws.

Most of the computation time in this three-step process lies in the first step, which currently requires a scan over all possible fault orientations. I improve the computational efficiency of this scan by using fast recursive smoothing filters for each orientation, but further improvements may be worthwhile. My current implementation of this scan for 500 fault orientations requires about two hours to process a 3D image of 1000^3 samples on a 12-core workstation.

I did not expect to find the conical shapes of faults apparent in Figures 1 and 2, in part because I had not recognized their hyperbolic appearance in horizontal and vertical slices of the 3D seismic image. An important benefit in using an automated process to extract faults from 3D seismic images is that the process cannot exclude such shapes simply because they are unexpected.

ACKNOWLEDGMENTS

In this work I benefited greatly from discussions with many others, including Luming Liang, Marko Maučec, Bob Howard, Dean Witte, and Anastasia Mironova. The 3D seismic image used in this study was graciously provided via OpendTect by dGB Earth Sciences B.V.

REFERENCES

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

Admasu, F., S. Back, and K. Toennies, 2006, Autotracking of faults on 3D seismic data: Geophysics, **71**, A49–A53.

Aqrawi, A., and T. Boe, 2011, Improved fault segmentation using a dip-guided and modified 3D sobel filter: Presented at the 81st Annual International Meeting, SEG, Expanded Abstracts.

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

Borgos, H., T. Skov, T. Randen, and L. Sonneland, 2003, Automated geometry extraction from 3D seismic data: Presented at the 73th Annual International Meeting, SEG, Expanded Abstracts.

Cohen, I., N. Coult, and A. Vassiliou, 2006, Detection and extraction of fault surfaces in 3D seismic data: Geophysics, **71**, P21–P27.

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

Kadlec, B., 2011, Visulation of geologic features using data representations thereof: US Patent Application 2011/0,115,787.

Kadlec, B., G. Dorn, H. Tufo, and D. Yuen, 2008, Interactive 3-D computation of fault surfaces using level sets: Visual Geoscience, **13**, 133–138.

Luo, S., and D. Hale, 2012, Unfaulting and unfolding 3D seismic images: CWP Report 722.

Marfurt, K., R. Kirlin, S. Farmer, and M. Bahorich, 1998, 3-D seismic attributes using a semblance-based coherency algorithm: Geophysics, **63**, P1150–P1165.

Marfurt, K., V. Sudhaker, A. Gersztenkorn, K. Crawford, and S. Nissen, 1999, Coherence calculations in the presense of structural dip: Geophysics, **64**, P104–111.

Pedersen, S., T. Randen, L. Sonneland, and O. Steen, 2002, Automatic 3D fault interpretation by artificial ants: Presented at the 72nd Annual International Meeting, SEG, Expanded Abstracts.

Pedersen, S., T. Skov, A. Hetlelid, P. Fayemendy, T. Randen, and L. Sonneland, 2003, New paradighm of fault interpretation: Presented at the 73nd Annual International Meeting, SEG, Expanded Abstracts.

Pishchulin, L., 2010, Matching algorithms for image recognition: Master's thesis, Rheinisch-Westfälischen Technischen HochSchule Aachen.

Sakoe, H., and S. Chiba, 1978, Dynamic programming algorithm optimization for spoken word recognition: IEEE Transactions on Acoustics, Speech, and Signal Processing, **26**, 43–49.

Schultz, T., H. Theisel, and H.-P. Seidel, 2010, Crease surfaces: from theory to extraction and application to diffusion tensor MRI: IEEE Transactions on Visualization and Computer Graphics, **16**, 109–119.

Taner, M., and F. Koehler, 1969, Velocity spectra — digital computer derivation and applications: Geophysics, **34**, 859–881.