3-D Semi-Automated Fault Interpretation in CASI Using Evolving Surfaces

Benjamin J. Kadlec, Geoffrey A. Dorn, Jonathon Marbach, Francis A. Coady

TerraSpark Geosciences

Abstract

An adaptable 3-D fault interpretation workflow that generates highly accurate surfaces is presented. The first step of the workflow generates an attribute that enhances fault discontinuities from a seismic volume. Next, seed points are generated either manually from a coarse handpicked fault region or automatically using a lineament extractor. An implicit surface is created from the seed points as an initial approximation to the faults. A fault segmentation algorithm then evolves the implicit surface, guided by the fault-enhanced attribute, into a dense planar representation of one or more faults. Classification techniques separate the planar representation into discrete surfaces based on dip and azimuth. The interpreter then uses an integrated 3D visualization technique to decide on a final interpretation by simple mouse strokes that merge the discrete surfaces into faults.

This technique addresses and overcomes limitations and pitfalls of similar methods. Results of this semi-automated approach are compared to a human interpretation in order to demonstrate accuracy and value.

Interactive techniques allow the interpreter to guide the generation of geologically meaningful faults that honor the structures represented in the seismic volume. Complex fault systems with intersecting and X-patterns are properly handled. The entire workflow is demonstrated as a case study in the Gulf of Mexico.

Overview

Fault Interpretation using evolving surfaces (also called level sets) works by evolving an initial surface according to a number of partial differential equations and attribute parameters. This surface can dynamically grow and shrink according to the fault surface. The process operates in three-dimensions on two inputs. The first input is a volume that enhances discontinuities in the dataset to highlight fault features. This volume defines the speed and direction by which the surface evolves and deforms. The representation is such that voxels with a high fault-likelihood value will grow and those with a low fault-likelihood value will shrink away and disappear. The second input defines a starting surface or initial seed points from which the faults begin evolving. This input can be in the form of a set of automatically extracted fault cuts, a manually interpreted fault cut, or an existing fault surface intended to be refined.

The process represents a fault as a "bounding surface" of the fault's 2-D manifold. This representation allows curvature to be defined at all points of the segmentation so that the actual fault surface can be segmented using a medial-surface extraction technique. The calculations for an evolving surface cannot be done on a simple 2-D manifold representing the fault.

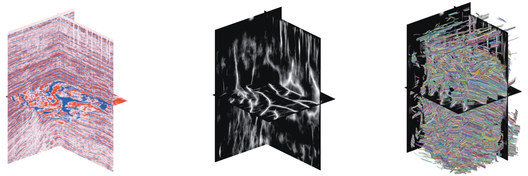

Figure 1: (Left) Input seismic volume. (Middle) Fault-Enhanced volume where bright voxels correspond to high-likelihood of a fault. (Right) Automatically extracted fault cuts approximate the location of faults and are the starting point for the evolving surfaces.

Three Workflows

Our approach is very adaptable and allows for a number of different input formulations, or workflows, to take advantage of the evolving surface representation. These workflows differ in the seed input given to our process. The first workflow, which we call **Automated Extraction**, uses as the input seed a set of automatically extracted fault cuts that approximate every fault in the volume. This workflow results in extracting a fault surface for all faults detected by the technique, as shown in Figure 2.

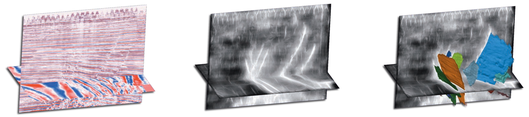

Figure 2: Automated Extraction requires no user input and uses automatically generate fault cuts of the entire volume as the seed input to our technique.

The second workflow, which we call **Semi-Automated Extraction**, uses as the input seed a manually interpreted fault cut. This fault cut or cuts can be a very coarse approximation of one or more faults that are intended to be interpreted. This workflow is designed as a follow-on to the fully automated approach to account for situations where a particular fault was missed by the automatic interpretation or was not segmented as desired as can be seen in Figure 3.

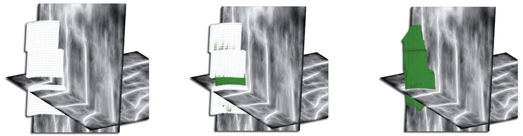

Figure 3: Semi-Automated Extraction requires manual interaction by the user to create coarse fault cuts that approximate the fault or faults which are desired to be extracted.

The third workflow, which we call **Semi-Automated Refinement**, uses as the input seed a previously interpreted fault surface, either picked manually or interpreted using this technique. There are often times when a fault surface has holes, missing information, or simply was not evolved far enough. In these situations the previously interpreted fault can be used as the seed input to our technique and evolved further in time to created a refined interpretation.

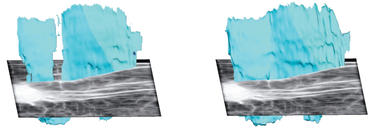

Figure 4: Semi-Automated Refinement requires a previously interpreted fault that needs to be refined to create a more continuous model of the surface.

Advantages

Truly 3-D Technique:
Many existing approaches to fault extraction operate at one point in their processing on 2-D slices of data from the volume. This is oftentimes done for purposes of computational speed and memory footprint, although there are disadvantages. In particular, working in 2-D can often lead to an inline or crossline bias in the fault interpretation which does not represent the underlying geology. Additionally, many faults in 3-D are not perpendicular to an inline or crossline slice and therefore can only truly be seen in 3-dimensions.

Complex Topologies Handled Implicitely:
The level set representation of an evolving surface that we use does not try and explicitly represent faults as triangulated surfaces until they are finished computing. This implicit representation allows fault surfaces to grow and merge or shrink and disappear in a way that is numerically sound and therefore the results are physically realizable, as is desired. Figure 5 compares our approach to a similar fault extraction technique that works explicitly by linking together fault cuts. It can be seen that faults with intersecting and complex structures are not easily handled explicitly.

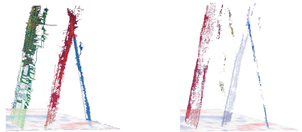

Figure 5: (Left) Incorrect interpretation using a vector linking approach does not handle crossing faults and contains spurious branches, while our evolving surfaces approach (right) handles the crossing fault topology implicitly.

Constrained Evolution:
In contrast to region growing techniques, our approach allows for a restrained and controlled surface evolution that prevents leaking and bleeding into non-fault regions. This is accomplished through the curvature term in the level set equation which enforces smooth evolution.

Simple Parameters:
Since our technique uses a fault-likelihood volume to determine fault locations, the primary parameter used in our approach is a threshold value that decides which voxel values are considered faults and which are considered background.

Dense and Precise Representation:
Our technique considers every voxel when evolving fault surfaces and this results in a dense set of points that represent a fault and requires little interpolation to see fine details of the surface. This is compared to other techniques that work with vectors of points and will often have a thin arrangement of points and "hairs" that hang off of the surface because they are incorrect picks.

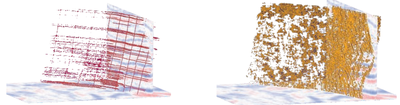

Figure 6: (Left) Imprecise interpretation of fault surface is represented by a sparse linking of vectors and fault detail is lost. (Right) Our evolving surfaces approach results in a dense set of points that precisely approximate the fault surface.

Future Work

As with most implicit modeling algorithms, the run time for our 3-D fault interpretation technique using evolving surfaces is much slower than other explicit techniques. The reason for this is simply due to the large number of derivatives that must be calculated over the entire domain at each iteration of the technique. Therefore, we will look into optimizing our current narrow-band implementation and consider implementations using massively parallel processing units, such as using a modern graphics processing unit (GPU).

Additionally, there are many improvements that can be made in the area of segmentation and clustering faults into coherent patches. Indeed, this is a large enough topic to justify its own research investigation and we only touch on the topic in this work. Integration of a clustering technique with an interactive user interface will allow geoscientists to further leverage their domain knowledge in grouping fault patches together.

Considering results, the faults output from this technique are not guarenteed to be continuous (ie: they may have holes and jagged edges). Although this may accurately represent the fault surface as it was imaged in the fault enhanced volume, it is often easier to work with a smooth and continuous fault surface. To solve this, we will look into triangulation techniques to in-fill fault holes and extend jagged edges without sacrificing accuracy in the results. Lastly, it should be stressed that fault evolution is not constrained by orientation, which means faults are allowed to "grow" in any direction such as perpendicular to the fault plane. There are situations where this is an undesired effect and can result in "fat" faults that grow into eachother. We can alleviate this problem by constraining fault evolution flow to occur only in the direction of the fault's medial surface. This is a topic of current research in our group.

References

G.A. Dorn, H.E. James, L. Evins, J. Marbach, F.A. Coady, Applications of Automatic Fault Extraction (AFE) in a Variety of Geologic Environments: Results, Limitations, and Suggested Improvements, AAPG Annual Meeting, 2007.

R. Fedkiw and S. Osher, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2002.

R. Kimmel, D. Shaked, N. Kiryati, A.M. Bruckstein, Skeletonization via distance maps and level sets, Computer Vision and Image Understanding, v. 62, n3, p. 382-391, 1995.

L. J. Latecki, Q. Li, X. Bai, W. Liu, Skeletonization using SSM of the Distance Transform, IEEE International Conference on Image Processing, 2007.

S. Osher and J.A. Sethian, Fronts Propagating with Curvature-Dependent Speed-Algorithms Based on Hamilton-Jacobi Formulations, Journal of Computational Physics, 1988.

D. Reniers, A. Telea, Skeletonization-based Hierarchical Shape Segmentation, IEEE International Conference on Shape Modeling and Applications, 2007.

M. Rumpf, A. Telea, A Continuous Skeletonization Method Based on LevelSets, Proceedings on the symposium on Data Visualization, 2002.

J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid