## Map complex fracture systems as termite mounds - a fast marching approach

Hao Guo\*, Kurt J. Marfurt<sup>#</sup> and Jiang Shu<sup>+</sup>

\* Hess Corporation, One Allen Center, 500 Dallas Street, Houston, Texas 77002

# ConocoPhillips School of Geology and Geophysics, University of Oklahoma, Norman, OK 73109-1009

+Department of Geological Sciences, University of Colorado, Boulder, CO 80309

#### Summary

Mapping complex fracture systems can be as difficult as mapping termite mounds. A zoologist casts a real 3D model of a termite mound by pumping 600-centigradealuminum melt into the tunnel system. We use a fast marching algorithm to simulate the casting process to help the interpretation of the complex geologic features such as fractures and collapse features. In our example, we find the algorithm to be fast, is amenable to user interaction, can handle complex geometry, and can implicitly handle bad picks.

#### Introduction

Rocks often form fracture systems as a result of the tectonic evolution. The tectonic history in the region controls the complexity of the fracture systems, which are sometimes quite complicated and intriguing as a termite mound to a zoologist.

If given a task to map a termite mound, a geoscientist has to test the seismic attributes responsive to the tunnel system and later to mentally prepare a structure model flexible enough to deal with the complex topology of the termite mound. The structure model is the most important guide for interpreting noisy or poorly imaged parts of the data. After weeks of painstaking picking, the prospect map is delivered to the file cabinet of the company with or without the accompanying interpretation model.

A zoologist will resolve this challenge with much more simplicity and brutality. He goes to the nest and pumps  $600^{0}$ C-aluminum melt into the nest. He waits a few hours for things to cool off and then digs the whole nest up along with the surrounding earth. After shaking off the dirt, an aluminum 3D model of the termite mound ends up in the natural science museum.

Let us save the mourning for the termite queen and her millions of children and focus on the physics of this mapping method. The aluminum melt is introduced into the system and the aluminum front evolves under the control of the permeability and interconnectivity of the tunnel. Please note that the zoologist has no priori structure model. When the process reaches equilibrium, the aluminum front preserves the complex 3D geometry of the termite mound. Inspired by the physics and appalled by the catastrophe in the termite mound, we want to adapt this mapping method digitally and find that the fast marching algorithm can do the trick and bring the inferno to the fracture systems. For those adept in constructing migration time tables using fast marching method, implanting the ideas in this paper simply requires changing a few lines in the code to read something other than velocity and adding one more GUIs to provide user interaction.

#### Theory and Method

Our workflow begins by reading in seismic attribute images/volumes suitable for fracture interpretation. Geoscientists then mark points, lines, or regions where they confidently interpret to be fractures. These marked features will serve as the entry points for the flood of "aluminum melt". The interpreters will supervise the flooding process and revise/terminate the process when they think necessary. For the numerical engine to pump the "aluminum melt" into the systems, we choose the fast marching method, a method widely used in seismic traveltime modeling and Kirchhoff migration

Fast marching method is a special type of level set method designed to track the evolution of interfaces (Sethian, 1987). One strength of the level set method is that it can simulate a complex N-dimension front using regular grids at an affordable price of doing computation in an N+1 dimensional domain. The fast marching method can simulate a non-contracting interface front (in other words, the velocity field for the propagating front is always nonnegative) at much faster speed than the general level set method. The most common application in seismic processing is to compute the traveltime table (Sethian and Popovici, 1999). The implementation detail of this method is referenced to Sethian and Popovici (1999) due to the size limit of this abstract.

Our proposed workflow has two modifications over the conventional construction of time tables using fast marching. First, we change the 3D input velocity volume to be an appropriate seismic attribute, for our problem volumetric curvature or coherence. Second, we are more interested in the *development* of the propagating font at

### Map complex fracture systems as ant nests – a fast marching approach

different stages instead of the time table itself. Since our intention is to adapt the fast marching algorithm to be an interpretation tool, we build a graphical user interface (GUI) to help users interactively supervise the front evolution and revise the input of the pseudo velocity.

We present a 2D case to illustrate our workflow. The 3D code is similar but not yet programmed.

## Examples

We select one composite image (Figure 6c from Guo et al., 2008) as our input image (Figure 1). Figure 1 is an overlay of a low coherence feature over a most negative curvature image. The circular low coherence features are interpreted to be collapse features while the black linear features correspond to high negative curvature features interpreted as faults and joints (Sullivan et al., 2006). We marked a few points (red arrows pointing to the locations) in the regions where we think collapse features exist. We also intentionally marked one bad point (the blue arrow pointing to the location) to test the algorithm's reaction to bad picks. The points marked by the red arrows and blue arrow serve as the entry points for the input of the "aluminum melt".

For the input velocity field, we tentatively set

$$V(x, y) = \frac{1}{1 + a^* I(x, y)}$$
(1)

Where V(x,y) is the velocity at , I(x,y) is the image value at (x,y) and *a* is chosen to be a constant in the case study. The basic idea is to make a low coherence/curvature event a fast conduit for the propagating "aluminum front". For fine-tuned interpretation, the geoscientist can define a new velocity function based on multiple attributes such as the azimuth of minimum curvature and the maximum curvature, while *a* can be generalized to be a function of (x,y) for finer control.

After setting up the input velocity, we begin to monitor the front evolution. Figures 2 through 5 show four sequential snapshots during this dynamic process. At first, the fronts grow rapidly inside of the collapse features (Figures 2 and 3). Later they begin to fill the collapse features and overflow through the connecting fracture systems (Figure 4). At the very end stage the isolated fracture systems begin to interconnect with each other as shown in Figure 5. Figure 5 shows a quite complicated zone made of interpreted fractures and collapse features, which is quite difficult to pick using lines and curves. Now with the assistance of the red lines marking interesting zones, the interpreters can relatively easily interpret the complex topology object into lineament, fracture and collapse feature components.

Please note that the isolated noise point marked by the blue arrow has never gotten the chance to grow and connect with other entry points marked by the red arrows. The result shows that the fast marching algorithm can shut down the supply of the "aluminum melt" to the inaccurate picks and thus implicitly contain their propagation for a long time. This observation reveals the eikonal nature of the fast marching algorithm, in that it simply calculates the traveltimes for all the neighboring points of the current front and the point with the shortest traveltime wins out. The new front will incorporate the new winning champion and go to the next iteration of competition until the time is up or until the new front hits the boundary of the image. This is the reason why the erroneous pick never get the supply of material to develop a significant feature during the stages shown in Figures 2 to 5. However, given sufficient a number of iterations, the erroneous pick will eventually spread and join with the more accurately picked fracture system thereby degrading the interpretation. At this point of our algorithm development, the geoscientist needs to monitor the progress of the progress of evolution, and back up if necessary, much as we routinely do with autopicking of seismic horizons.

#### **Discussions and Conclusions**

Our example shows that the well-established fast marching algorithm is capable of mapping complex geologic features such as fracture zones and collapse features. Its ability to handle complex geometry enables the algorithm to map features that are otherwise difficult to visualize and interpret. Its speed makes it amenable to interactive exploratory data analysis by interpreters who wish to evaluate alternative hypotheses. Its basis on the eikonal equation allows it to handle the errors inadvertently brought by interpreters.

The interpreters are in control of this tool by changing the input velocity field, picking different starting point assembly and controlling the time of fast marching process. The most important parameter is the attribute (or combination of attributes) v(x,y) used. Selection of the appropriate attribute (along with the definition of a(x,y)) requires a good understanding of seismic geomorphology, diagenesis, and tectonic deformation, as well as the response of seismic data to underlying rock properties. The resulting traveltime value can be interpreted to be a statistical measure of confidence with smaller traveltime implying higher confidence or similarity to the input points. In this manner, these results can serve as input to subsequent geostastical analysis.

# Map complex fracture systems as ant nests - a fast marching approach



Figure 1: The input of coherence-curvature composite image to the 2D fast marching algorithm. The arrows point to the entry points of the "aluminum melt" flood. The red arrows indicate proper interpretations and the blue arrow indicates an erroneous pick intensionally left to test the robustness of the program.



Figure 3: The fronts develop at a premature stage at iteration 2. Some fronts already show the shapes of the collapse features while others are still developing.



Figure 2: The fronts develop at an embryonic stage at iteration 1. The fronts develop inside some of the collapse features



Figure 4: The fronts develop at a mature stage at iteration 3. The fronts cover the outer rim of all the identified collapse features.

## Map complex fracture systems as ant nests - a fast marching approach



Figure 5: The fronts develop at a post-mature stage at iteration 4. The fronts overflow the identified collapse features along the complex fractures networks.

#### AKNOWLEDGEMENTS

Thanks to Devon Energy for providing access to the seismic data to UH (and later OU) for use in research and education.

#### **REFERENCE:**

Brown M.P., Scott A.M., and W. Gary, 2006, Seismic event tracking by global path optimization, SEG Expanded Abstracts **25**, 1063-1067.

Guo H., S. Lewis, and K. J. Marfurt, 2008, Mapping multiple attributes to three-and four-component color models - a tutorial: Geophysics, **73**, 7-19.

Sethian, J. A., 1985, Numerical methods for propagating fronts: in Variational methods for free surface interfaces, *in* Proceedings of the September, 1985 Vallambrosa Conference, Eds. P. Concus and R. Finn: Springer-Verlag, NY, 1987.

Sethian, J. A. and M. A. Popovici, 1999, , 3-D traveltime computation using a fast marching method: Geophysics, **64**, 516-523.

Sullivan, E. C., K. J. Marfurt, A. Lacazette, and M. Ammerman, 2006, Application of new seismic attributes to collapse chimneys in the Fort Worth Basin: Geophysics, **71**, 111-119.