Atomic meshing of seismic images

Dave Hale and Jeff Emanuel, Landmark Graphics*

Summary

Today's work cycle from seismic imaging to reservoir simulation requires a variety of data structures - simple arrays, triangulated surfaces, non-manifold frameworks, corner-point grids, etc. - to represent the earth's subsurface. Conversions among these different representations are both time consuming and error prone.

Using simple image processing techniques, we automatically align a lattice of points (atoms) with horizons and faults in a seismic image. Connecting these points yields an unstructured space-filling polyhedral (atomic) mesh. This single data structure can integrate multiple tasks, such as seismic interpretation, reservoir characterization, and flow simulation, thereby reducing work cycle times and errors.

Introduction

One often analyzes seismic images to obtain meshes that facilitate further computation. For example, consider the horizontal slice of a 3-D seismic image of geologic faults shown in Figure 1. Figure 2 shows a space-filling mesh of triangular elements that has been aligned with those faults (bold white lines). Flow vectors (black line segments) indicate both the direction and velocity of fluid flowing from an injector in the upper left to a producer in the lower right part of the image. *We performed both the fault interpretation and the flow simulation on the same space-filling mesh.*

As in this example, the mesh used for petroleum reservoir simulation is typically not the uniform grid on which seismic data are sampled and processed. To reduce computation costs, a flow simulation mesh may be sampled more coarsely, especially along horizontal dimensions, and its sampling may not be uniform. To better conform to subsurface geology, the reservoir simulation mesh may also be unstructured — connections among mesh elements may be explicit, and not simply implied by array indices. As reservoir simulation becomes more widely used, seismic images are increasingly analyzed to create meshes suitable for this and other purposes.

Today such analyses often consist of the following sequence of steps (e.g., Garrett et al., 1997):

(1) Process an image to enhance features of interest.

(2) Find curves or surfaces in the image that bound regions of interest.

(3) Fill the space defined by those regions with a mesh.

(4) Simulate some process on the space-filling mesh.

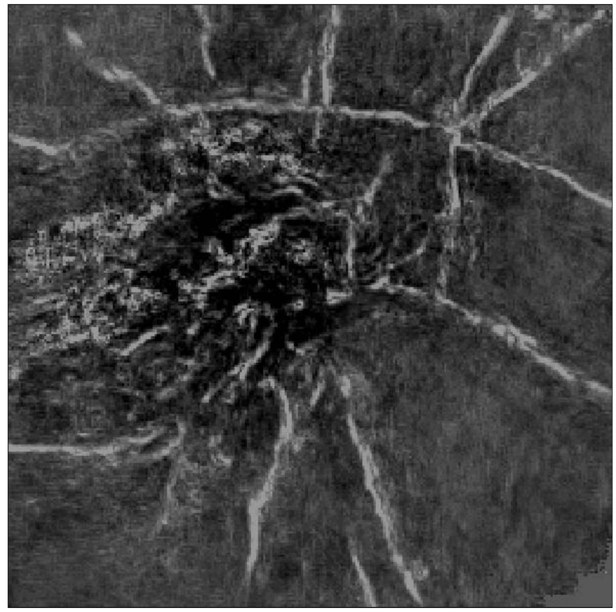

FIG. 1: A seismic image of geologic faults. This 256×256-sample (6.4×6.4-km) 2-D image is a horizontal slice taken from a 3-D seismic image that was processed to enhance such discontinuities.

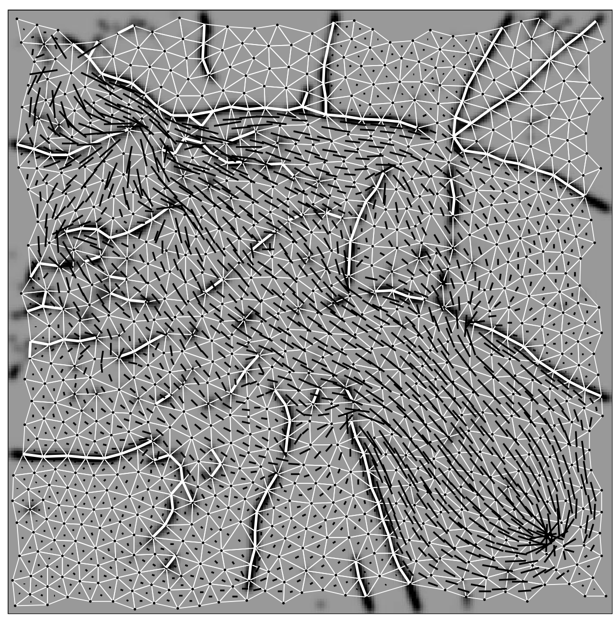

FIG. 2: This space-filling mesh was automatically aligned with features in the seismic image, and then used to identify geologic faults and simulate the flow of fluid around them.

For various reasons, each step in this sequence is today often taken independently, with little regard for the requirements of subsequent steps. For example, it is common in step (2) to produce a bounding curve or surface with more detail than can be represented in the space-filling mesh used in step (4).

This discrepancy in resolution is often accompanied by a discrepancy in data structure. For example, a two-dimensional (2-D) curve represented by a simple linked list of line segments in step (2) may become a relatively complex mesh of triangles in step (3). Such discrepancies today disrupt the analysis sequence, and may yield inconsistencies between the image processed in step (1) and the mesh used in step (4) that are difficult to quantify.

The disruptions are costly. In the example of seismic image analysis and reservoir simulation, one iteration of this sequence today may require a month or more of work. This high cost makes it difficult to perform multiple iterations in attempts to estimate uncertainties in simulation results.

In the computations leading to Figure 2, we used a more direct sequence:

(1) Process an image to enhance features of interest.

(2) *Fill space with a mesh aligned with image features.*

(3) Simulate some process on the space-filling mesh.

Instead of finding boundaries of regions within seismic images and then meshing those regions, one simply constructs a mesh that is aligned with the boundaries.

In this paper, we focus primarily on step (2), which we call atomic meshing.

Atomic meshing of seismic images

In the atomic mesh shown in Figure 2, the atoms lie at the vertices of the triangular mesh elements. These vertices are often called the "nodes" of the mesh. Here, we call them "atoms", because we create the mesh using simple models of inter-atomic forces. Atoms separated by less than some nominal distance repel each other. Atoms separated by more than that nominal distance attract each other. These attractive and repulsive forces cause randomly distributed atoms to arrange themselves in a geometrically regular lattice, as in a crystal.

Crystal lattices provide a useful analogy for geological modeling. In this analogy, layers of atoms in crystals correspond to geologic layers, and dislocations in crystals correspond to geologic faults. We manipulate geologic models by moving atoms. As we move one atom, atoms nearby will move along with it, thereby preserving the regularity of the lattice. Physical models such as this are common in meshing; see Shimada (1993) for a most relevant example.

Building on this analogy, we computed the space-filling mesh shown in Figure 2 using the following process (Hale, 2001):

(1) Fill the space spanned by the image with a pseudo-random lattice of atoms.

(2) Move the atoms to minimize a total potential energy, defined to be a weighted sum of an atomic potential energy and an image potential energy.

(3) Connect the atom locations via Delaunay triangulation to form a mesh.

Step (2) of this process moves atoms automatically, so that they are well aligned with image features while otherwise tending towards a geometrically regular lattice. We computed the dark features in the image of Figure 2 from the seismic image in Figure 1, using well-known image processing techniques for detecting such quasi-linear features. In step (2), these features exert an attractive force on the atoms. Simultaneously, repulsive and attractive forces among atoms causes them to maintain a nominal distance to their nearest neighbors. The balance of these image and atomic forces yields the lattice and mesh shown in Figure 2.

It may be useful to think of the image features (the faults) in Figure 2 as potential valleys. Atoms tend to roll into the valleys, while inter-atomic forces maintain their separation.

Step (2) is the most computationally demanding, as it requires repeated computation of the total potential energy and its partial derivatives with respect to the spatial coordinates of each atom. In our context of meshing an image, we use well-known image processing techniques (e.g., convolution and finite-differencing) to compute these quantities, which we then provide to a generic algorithm for minimizing the potential energy function of many atom locations (Hale, 2001).

That minimization algorithm is iterative; it moves atoms repeatedly in its search for a minimum potential energy. Therefore, for computational efficiency, it is significant that atoms are first connected to form a mesh only in the last step (3). If atoms were connected before optimizing their locations, most computation time during optimization would be spent breaking and remaking connections as atoms are moved.

After using this process to align the mesh, we may edit the mesh by adding, removing, or moving atoms, as necessary. This editing may be performed interactively, as we repeat step (2) of the process above for only those atoms near the region being edited. During editing, we simultaneously update the mesh by breaking and remaking connections among atoms.

The sizes of mesh elements (e.g., triangles) computed in step (3) depend on the density of atoms in the initial pseudo-random lattice of step (1). That atomic density may vary spatially, consistent with the density of features in the image. This ability to sample finely where detail is warranted, while sampling more coarsely where it is not, is one of the desirable features of an unstructured mesh. We illustrate this feature in the examples below.

More examples

We computed the triangular mesh in Figure 2 by automatically aligning atoms *on* image features. Alternatively, we may align atoms *alongside* image features, as in the polygonal mesh shown in Figure 3.

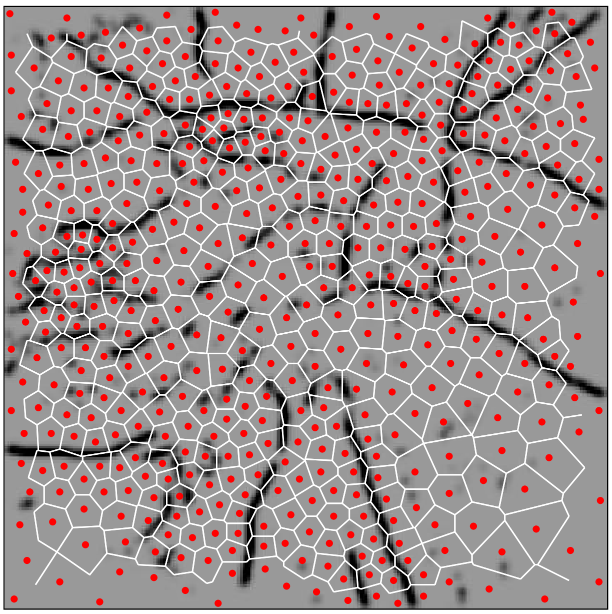

FIG. 3: A polygonal space-filling mesh aligned with the faults imaged in Figure 1.

The polygonal mesh elements have edges coincident with image features; a consequence of aligning atoms alongside those features. Note that the atoms lie within the polygons, not at their vertices. Also, note that mesh elements near the middle left part of the image are smaller than those near the lower right part, consistent with the variable density of image features.

The three-step process used to compute this mesh is essentially the same as that described above. The only difference is that, here, we let image features (the faults) repel atoms instead of attracting them. Whereas the features in Figure 2 represent potential valleys, the features in Figure 3 represent potential hills; atoms tend to roll away from them. They do not roll far away, however, because of attractive and repulsive inter-atomic forces.

In all other aspects, the processes used to obtain the triangular or polygonal meshes are the same. In particular, the Voronoi polygons in Figure 3 correspond to a Delaunay triangulation that we compute in step (3) of that process. We do not display this triangulation, because the edges of the triangles are not coincident with image features.

Figure 4 shows a polygonal mesh that we aligned with a combination of both horizons and faults in vertical slices extracted from two 3-D seismic images. One image was processed to enhance horizons, the other to enhance faults. We combine both images in this slice, before atomic meshing, so that the mesh is aligned with both types of features. This alignment enables properties that we may associate with mesh elements (e.g., permeability) to be continuous between horizons but discontinuous across faults.

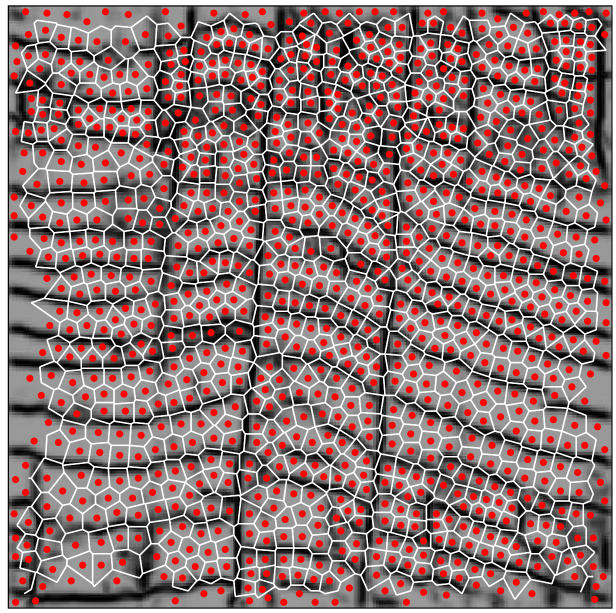

FIG. 4: A polygonal space-filling mesh aligned with a combination of both horizons and faults in vertical (256×256-sample, 6.4×0.8-km) slices extracted from two 3-D seismic images.

We find Voronoi polygonal meshes, such as those in Figures 3 and 4, preferable to Delaunay triangular meshes, for several reasons. First, Voronoi polygons vary smoothly with small perturbations of atom locations, whereas Delaunay triangles do not. In a Delaunay triangulation, a slight change in atom location may cause existing triangles to be destroyed and new triangles to be created. Second, properties that we associate with Voronoi polygons (e.g., permeability) are implicitly associated with atoms, and those properties remain well-defined as we move atoms. In contrast, properties associated with Delaunay triangles may become undefined as triangles are destroyed and created. Third, in a 3-D atomic mesh, there are about six times as many Delaunay tetrahedra as there are Voronoi polyhedra. Finally, 3-D Voronoi polyhedra tend to be well shaped, whereas some 3-D Delaunay tetrahedra (called "slivers") tend to be poorly shaped, with large surface area but small volume.

Figures 5 and 6 illustrate atomic meshing and interpretation of a 3-D seismic image. The polygons displayed in the three orthogonal slices in Figure 6 are slices of a *3-D polyhedral space-filling mesh*. (Slicing one polyhedron yields one polygon.) As in the 2-D example above, the mesh was aligned automatically with a combination of both horizons and faults from two 3-D seismic images.

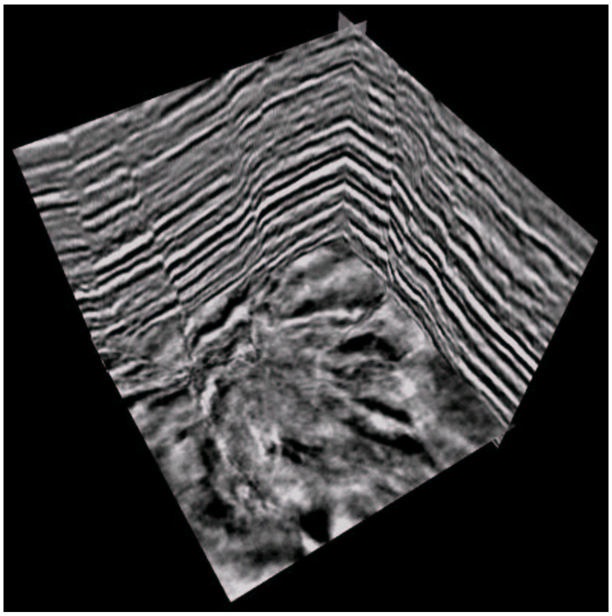

FIG. 5: Three orthogonal slices from a $256 \times 256 \times 256$-sample ($6.4 \times 6.4 \times 0.8$-km) 3-D seismic image.

After meshing, we interpreted the 3-D seismic image by interactively painting polyhedra, so that atoms within the same interpreted unit have the same color. (The interpretation is best viewed in the full-color electronic version of this abstract.) We then selected one unit painted gold and computed the surface surrounding all atoms with that color. Parts of that surface are shown in Figure 6. Large displacements have caused cause several fault blocks for this unit to become disconnected.

Interpretation by painting an atomic mesh is much more efficient than painting individual samples of a 3-D seismic image. In this example, the average polyhedron contains roughly 300 samples of the 3-D seismic image. Also, before painting, we automatically computed clusters, each containing no fewer than 15 polyhedra. Therefore, painting clusters of polyhedra was at least 4500 times more efficient than painting samples of the 3-D seismic image.

Conclusions

Atomic meshing fills the space spanned by a 3-D image with a mesh that is automatically aligned with features in that image. In the words of Sword and Toldi (1995), we "treat the region [to be meshed] as a solid, rather than a void, and squish it around." We therefore bypass

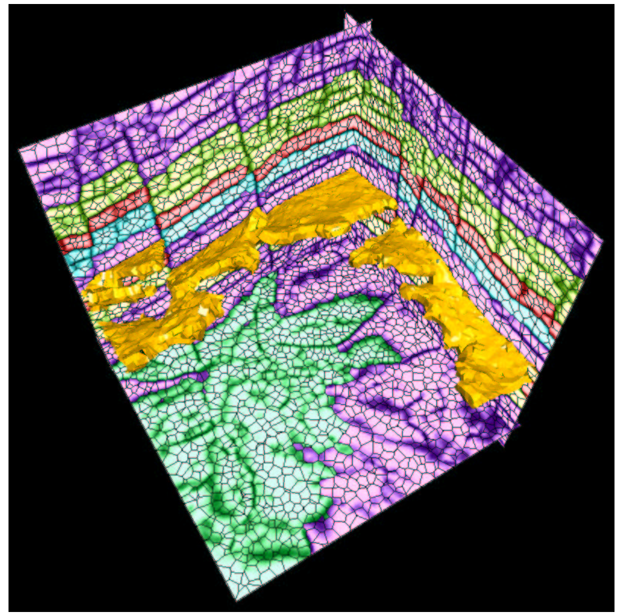

FIG. 6: Interpretation of the image of Figure 5 after 3-D atomic meshing. (See the full-color electronic version.)

problems, such as topological consistency, that arise when "defining regions of space by a set of bounding surfaces."

While the ultimate goal of this work is to enhance the integration of tasks such as seismic interpretation and reservoir simulation, we find that atomic meshes may enhance 3-D seismic interpretation alone. In particular, interactive painting of 3-D seismic images becomes feasible.

References

Garrett, S., Griesbach, S., Johnson, D., Jones, D., Lo, M., Orr, W., and Sword, C., 1997, Earth Model Synthesis, First Break, v. 15, no. 1, p. 13–20.

Hale, D., 2001, Atomic images - a method for meshing digital images: Proceedings of the 10th International Meshing Roundtable, p. 185–196.

Shimada, K., 1993, Physically-Based Mesh Generation: Automated Triangulation of Surfaces and Volumes via Bubble Packing, Ph.D. thesis, Massachusetts Institute of Technology.

Sword, C. and Toldi, J., 1995, 3-D model representations: poster paper presented at the 65th Ann. Internat. Mtg., Soc. Explor. Geophys.

Acknowledgements

Thanks to Bob Howard for asking really good questions, and to John Killough and Bill Watts for insights on flow simulation.