

## Seismic interpretation using global image segmentation

Dave Hale\* and Jeff Emanuel, Landmark Graphics

### Summary

A first step in seismic interpretation is seismic image segmentation. For most seismic images, with incompletely or poorly imaged faults and horizons, global methods for segmentation are more robust than local event tracking or region growing methods commonly used today.

The disadvantage of global image segmentation methods has been their relatively high computational cost. We reduce this cost by applying these methods to a space-filling mesh aligned automatically with features in seismic images. The mesh makes global segmentation of 3-D seismic images feasible.

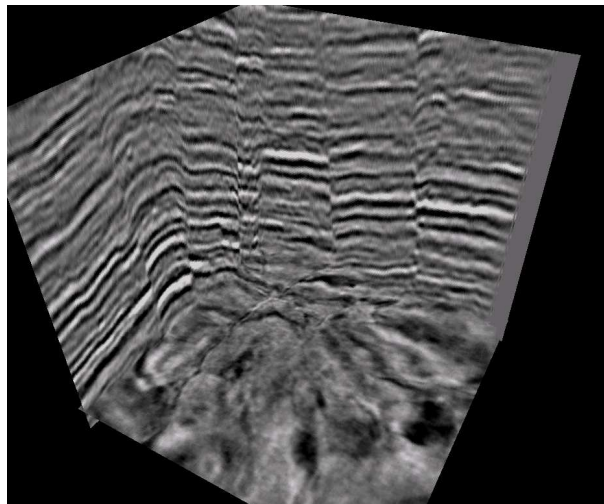
### Introduction

An important step in seismic interpretation is segmentation of a seismic image into structural and stratigraphic geologic units. Today, we often perform this segmentation using a simple process. We first automatically track seismic events to locate horizons and faults that correspond to the boundaries of geologic units. We then glue those boundaries together to obtain a geologic model. This process is equivalent to segmentation of a seismic image.

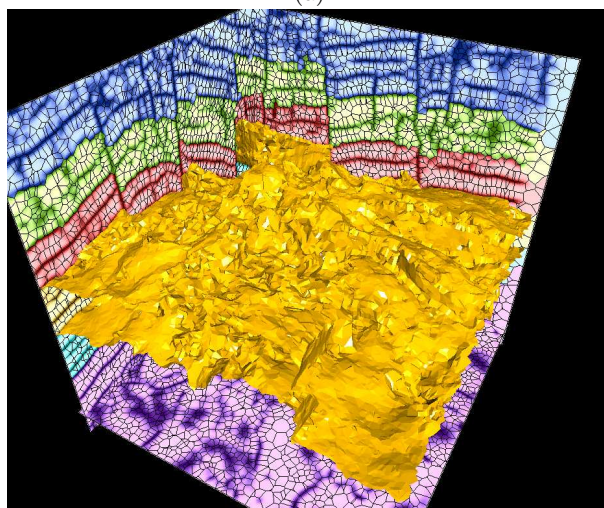
Figure 1 illustrates an alternative process for seismic image segmentation. In this process, we first align a space-filling polyhedral mesh with features in a 3-D seismic image (Hale, 2002; Hale and Emanuel, 2002). In this example, a typical mesh element contains approximately 160 image samples. We then automatically segment the mesh and, hence, the image into clusters (not shown) of contiguous polyhedral elements. In this example, each cluster contains approximately 60 polyhedra, about 10,000 image samples. We interactively painted those clusters to obtain the six geologic units shown in Figure 1b.

This new alternative process for seismic image segmentation is interesting, because the simple process used today is, in practice, not so simple. Seismic data often lack the fidelity necessary for event trackers to yield error-free unit boundaries that can be simply glued together. Typically, we must edit, extrapolate, and truncate those boundaries as we combine them to form a topologically (and therefore geologically) consistent model.

Most automatic seismic event tracking methods work by solving a sequence of *local* optimization problems. Beginning with one or more seed samples, a seismic event tracker looks for adjacent samples that are most similar. The tracker then adds those samples to the set of seeds, and repeats. While some implementations may be more robust than others, event tracking algorithms share a common flaw. They are near-sighted. They make only locally optimal choices, like a novice chess player that can



(a)



(b)

FIG. 1: Three orthogonal slices from a  $256 \times 256 \times 256$ -sample ( $6.4 \times 6.4 \times 0.8$ -km) 3-D seismic image (a) before and (b) after seismic interpretation. Interactive painting of the six highly-faulted geologic layers shown here was facilitated by automatic segmentation of a space-filling polyhedral mesh. Slices of that mesh are also shown here. Two yellow surfaces bound the layer painted yellow. (See the color electronic version.)

see only one or two moves ahead.

In contrast, the segmentation method used to obtain Figure 1b works like a chess player that can see the whole board. It solves a *global* optimization problem, and is a

## Seismic interpretation using global image segmentation

variant of the normalized cuts method (Shi and Malik, 1997) for image segmentation, well known in the field of computer vision.

Although well known for their robustness, we are not aware of global methods such as this being used for seismic image segmentation. They are probably too expensive, requiring too much computation time. Published examples of their application in computer vision typically show 2-D images with fewer than 100,000 samples. 3-D seismic images may have 1,000,000,000 samples.

In our variants of such global methods for image segmentation, we reduce their cost significantly, by applying them to mesh elements aligned with image features, instead of individual image samples.

### Meshed images

For clarity below, we illustrate global methods for seismic image segmentation with a single 2-D image. The image in Figure 2a is a 2-D horizontal slice from the 3-D image of Figure 1a, after processing to enhance faults. Figure 2b illustrates the alignment of a space-filling polygonal mesh with those faults.

We performed this alignment using the *atomic meshing* process described by Hale (2002) and Hale and Emanuel (2002). In those publications, we highlight the use of such meshes in flow simulation. Here, we demonstrate their use in seismic image segmentation.

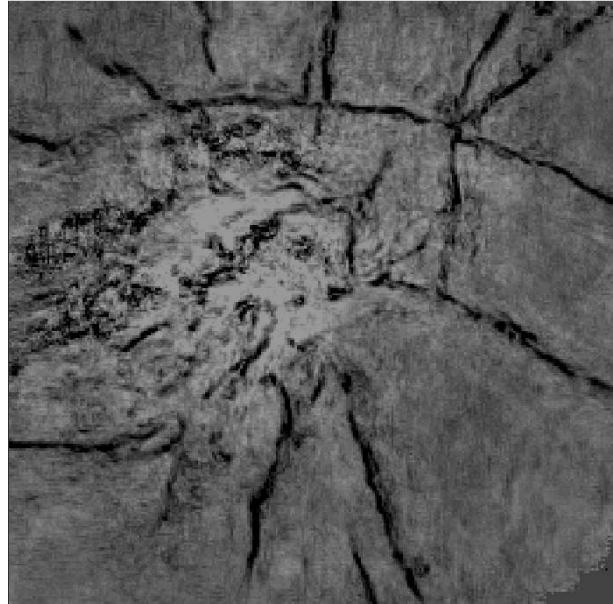
Note that the mesh samples the image more finely (with smaller polygons) where image features are dense, and more coarsely where they are sparse. Whereas the image contains 65,536 samples, the mesh contains only 1348 polygonal elements. In our adaptations of global methods for image segmentation, we process each of those mesh elements as if it were a single image sample.

For example, both of the segmentation methods we consider here begin by computing a weight for each pair of adjacent image samples. To reduce costs, in our implementations, we compute a weight for each pair of adjacent mesh elements. The weights are simply numbers between zero and one. Where adjacent mesh elements lie on opposite sides of an image feature (here, corresponding to a fault), the weights are small. Elsewhere, where no faults are imaged, the weights are large. Intuitively, we expect segment boundaries to be created where weights are small.

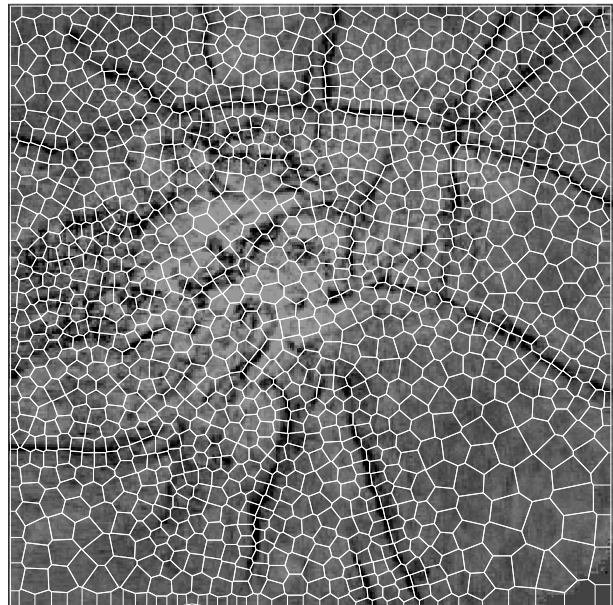
### Normalized cuts

In the normalized cuts method, we use the weights computed from the mesh and the image to construct a large sparse eigenproblem. The matrix for that eigenproblem is similar to the matrix one obtains for the simplest solutions to the pressure equation in flow simulation, if we assume that faults represent barriers to fluid flow. Indeed, the numerical methods we use to solve the eigenproblem are very similar to those used in flow simulation.

As described by Shi and Malik (1997), we use the first few



(a)



(b)

FIG. 2: A  $256 \times 256$ -sample ( $6.4 \times 6.4$ -km) horizontal slice (a) from a 3-D seismic image processed to enhance faults and a space-filling polygonal mesh (b) aligned with features in that image.

eigenvectors of this matrix, corresponding to its smallest eigenvalues, to cut the mesh into two or more parts. Each cut has a cost, defined by the normalized cut criterion. We then recursively cut each of those parts until either the cost of a cut is greater than a specified upper bound or until the number of mesh elements in a part is less than a

## Seismic interpretation using global image segmentation

specified lower bound. When each cut splits the mesh into two parts, this recursion yields a (typically, unbalanced) binary tree of parts.

Figure 3a is the result of cutting the mesh once into two parts. Figure 3b is the result of cutting those two parts, recursively, to obtain twenty parts. Note that cutting the mesh is equivalent to segmentation of the image.

Note also that segment boundaries are reasonable even where faults are not well imaged. This robustness is the most compelling feature of a global segmentation method. In contrast, a local tracking or region growing method would tend to leak through these holes in the image.

### Stochastic clusters

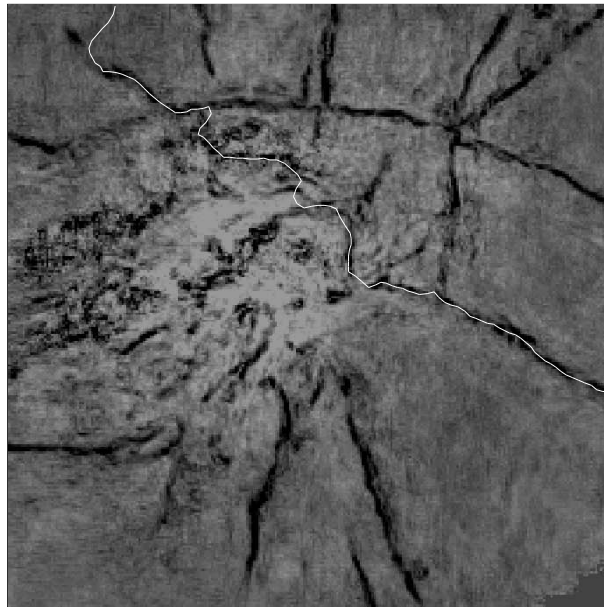
In contrast to the normalized cuts method, which recursively splits larger parts of the image into smaller parts, the stochastic clusters method iteratively combines smaller parts of the image into larger parts. As described by Gdalyahu et al. (2001), this method begins with one part for each image sample, with weights computed for each pair of adjacent samples. In our implementation, we begin with one part for each mesh element, with weights computed for each pair of adjacent elements.

Conceptually, the stochastic clusters method works as follows. We distribute the mesh to hundreds or even thousands of seismic interpreters. We ask all interpreters to *randomly* choose and combine one pair of adjacent mesh elements, with probability proportional to the weight we computed for that pair. Then, for each pair that could have been chosen, we take a poll. If more than half the interpreters have chosen a particular pair, we combine those mesh elements. We then repeat this process, until all mesh elements have been combined.

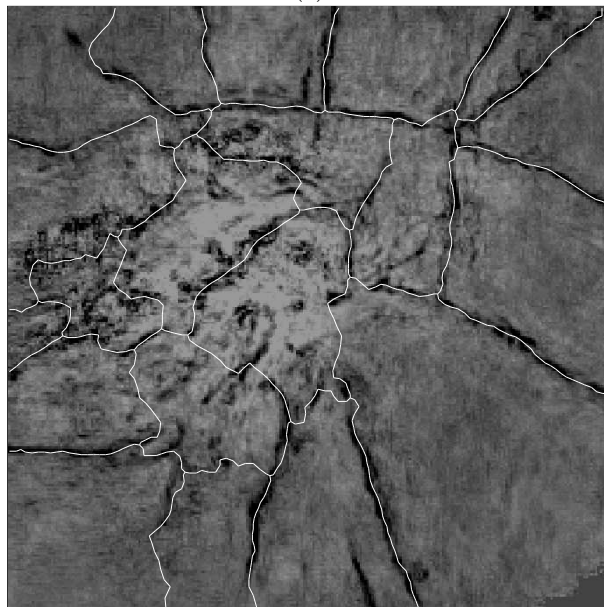
If we begin with  $N$  mesh elements, then we can repeat this process  $N - 1$  times. At early times, it is highly unlikely that half of our interpreters will have chosen the same pair of adjacent elements to combine. There are simply too many good choices, pairs with weights close to one. However, at late times, the odds of a majority vote increase significantly, because we count all votes cast by each interpreter at earlier times. And at certain times in between, clusters of elements will form, like crystals, as in an annealing process, as if we were decreasing temperature with time.

Figure 4 illustrates the stochastic clustering process at early and late times. At early times, when temperatures are high, clusters are small. Later, as temperature decreases, the clusters become larger. In this example, we have chosen a later time that yields a segmentation similar to the one we obtained using the normalized cuts method. Compare Figures 3b and 4b.

Of course, the interpreter in our conceptual description of stochastic clustering is actually a very simple computer algorithm. It makes choices using only a pseudo-random number generator and the weights for pairs of adjacent mesh elements. Its simplicity is the reason that we require



(a)



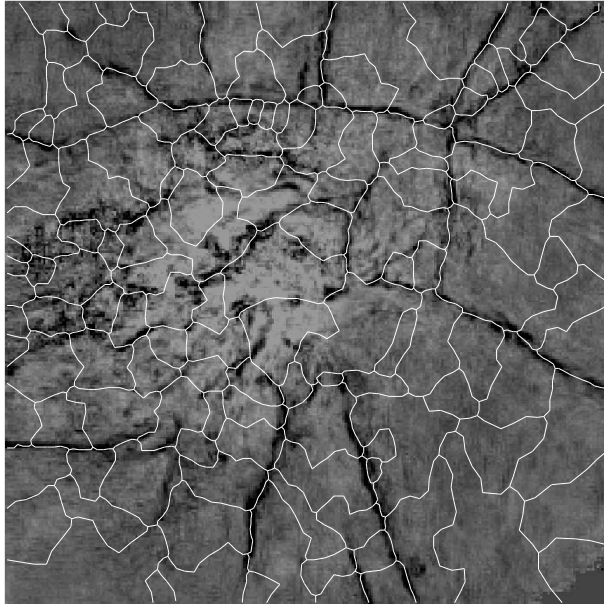
(b)

FIG. 3: Seismic image segmentation using a normalized cuts method. We first make one cut, to obtain (a) two parts. We then recursively cut those parts to obtain (b) 20 parts.

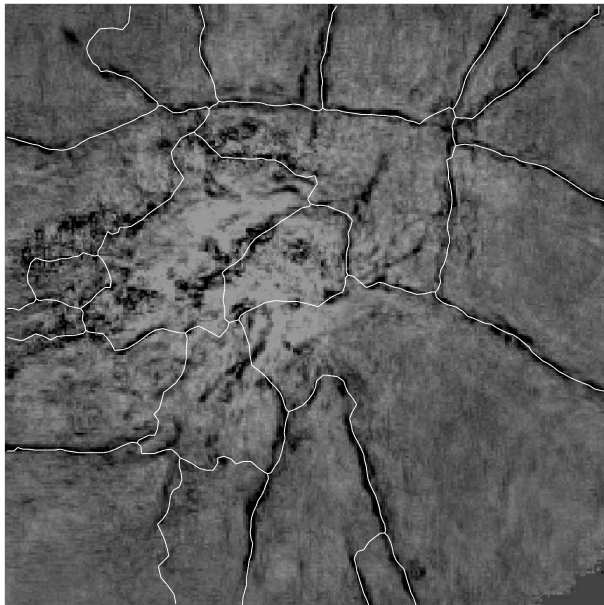
a majority vote before we combine mesh elements.

Although each of our stochastic interpreters is not very clever, each works with the entire mesh. The pairs of mesh elements that an interpreter chooses at one time and the next may be far apart. Unlike a human interpreter that tends to work within one or two units visible in one

## Seismic interpretation using global image segmentation



(a)



(b)

FIG. 4: Seismic image segmentation using a stochastic clusters method. At an early time (high temperature), we have (a) 145 clusters. At a later time (lower temperature), those clusters have merged to form (b) 20 clusters. Compare with Figure 3.

2-D image slice at a time, our simple interpreter skips around the 3-D meshed image, randomly putting parts together. This global behavior and the collective wisdom of many interpreters yields a segmentation that is robust with respect to poorly imaged features.

### Conclusion

The normalized cuts and stochastic clusters methods for image segmentation are global. Unlike local event tracking or region growing methods, these methods work with the entire image or, in our implementations, the entire *meshed* image. Both methods yield a hierarchy of parts that we use to interactively split or combine geologic units during seismic interpretation.

The two methods construct that hierarchy of parts in opposite ways. The normalized cuts method begins with one large part and recursively splits the mesh into smaller parts. The stochastic clusters method begins with many small parts (mesh elements) and combines them into larger parts.

In our experiments, including tests not shown here, the normalized cuts and stochastic clusters methods yield similar results. Currently, we favor the normalized cuts method, because our implementation of that method on single processor systems is roughly four to eight times faster and requires less computer memory than the stochastic clusters method. However, the latter method can more easily exploit multi-processor systems.

Neither segmentation method completely automates seismic interpretation. For example, neither method automatically correlates across discontinuous fault blocks. Today, we complete seismic interpretations by interactively combining and painting segments that we obtain using the normalized cuts method.

Nevertheless, we speculate that further automation is possible. A common but seldom achieved goal in computer vision is to match the accuracy of the human visual system. In processing 2-D images, a human is almost always more accurate. However, for 3-D images, humans may lose. We tend to focus on one 2-D slice at a time. We forget the big picture. We do not see the whole board. Automated global methods for 3-D seismic interpretation need not suffer these shortcomings.

### References

- Gdalyahu, Y., Weinshall, D., and Werman, M., 2001, Self organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization: IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 23, no. 10, p. 1053–1074.
- Hale, D., 2002, Atomic meshes - from seismic imaging to reservoir simulation: Proceedings of the 8th European Conference on the Mathematics of Oil Recovery.
- Hale, D., and Emanuel, J., 2002, Atomic meshes of seismic images: presented at the 72nd Ann. Internat. Mtg., Soc. Explor. Geophys.

Shi, J., and Malik, J., 1997, Normalized cuts and image segmentation: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, p. 731–737.