Image segmentation for tracking salt boundaries

Jesse Lomask, Biondo Biondi, and Jeff Shragge

SUMMARY

Image segmentation can be used to track salt boundaries when the salt boundary amplitude is greater than any other local reflections. We apply a modified version of the normalized cut image segmentation method to partition seismic images along salt boundaries. In principle our method should work even when the boundaries are not continuous, and conventional horizon tracking algorithms may fail. Our implementation of this method calculates a weight connecting each pixel in the image to each pixel in a local neighborhood. The weight is made weak where the negative amplitude of the complex trace along the shortest path between the two pixels has a minimum and is less than a threshold value. This method is demonstrated to be effective on synthetic 2D seismic sections and can easily be modified to be applied to 3D data. To overcome the formidable computational expense and storage requirements, three cost saving approaches are proposed. Firstly, pixels are sampled from windows centered at powers of 2, this greatly increases the sparseness of the weight matrix. Secondly, initial solutions are provided to subsequent segmentations for multiple segmentation passes in iterative velocity analysis. Thirdly, an iterative multiscale approach should allow the tracking of the bright salt events in large 3D cubes.

INTRODUCTION

Salt boundaries are often the brightest, most prominent reflections in a seismic image. For many processing applications, this boundary needs to be tracked. However, this reflection can be discontinuous and have significant amplitude variation, making it difficult for traditional amplitude based auto-pickers to track. Other tracking techniques such as artificial neural networks are less sensitive to amplitude variations but are still prone to error if the seismic wavelet character varies significantly from the training data (Leggett et al., 1996). Here, we present a tracking method that can handle these issues.

Hale and Emanuel (2003, 2002) applied the normalized cut image segmentation method developed by Shi and Malik (2000) to paint a 3D coherency based reservoir model. Our approach is very similar. This image segmentation technique creates a matrix containing weights relating each pixel to every other pixel in a local neighborhood. The matrix is then used to cut the image where the normalized sum of weights cut is minimized. We have modified the weight calculation to be dependant on the negative absolute value of the complex trace (instantaneous amplitude) of the seismic. This makes the weights very weak at salt boundaries, causing the segmentation algorithm to cut along the boundary.

In this paper, we give a very general overview of the normalized cut segmentation technique. We then describe how we modified it for application to salt dome seismic data. We test this technique on some synthetic seismic sections illustrating its efficacy with discontinuous salt boundaries. Approaches to make this method more robust and cost effective are also presented.

SEGMENTATION METHODOLOGY

The normalized segmentation method described by Shi and Malik (2000) is designed to look for clusters of pixels with similar intensity. To do this, it first creates a weight matrix relating each pixel to every other pixel within a local neighborhood. The strongest weights will be between pixels of similar intensity and close proximity to each other. The method then seeks to partition the image into two groups, A and B, by minimizing the normalized cut:

$$\mathrm{N_{cut}} = \frac{\mathrm{cut}}{\mathrm{total_A}} + \frac{\mathrm{cut}}{\mathrm{total_B}} \tag{1}$$

where cut is the sum of the weights cut by the partition. $total_A$ is the sum of all weights in Group A, and $total_B$ is the sum of all weights in Group B. Normalizing the cut by the sum of all the weights in each group prevents the partition from selecting overly-small groups of nodes.

The minimum of N_{cut} can be found by solving the generalized eigensystem:

$$(\mathbf{D}-\mathbf{W})\mathbf{y} = \lambda\mathbf{D}\mathbf{y} \tag{2}$$

created from the weight matrix ($\mathbf{W}$) and a diagonal matrix ($\mathbf{D}$), with each value on the diagonal being the sum of each column of $\mathbf{W}$. The eigenvector with the second smallest eigenvalue is used to partition the image by taking all values greater than zero to be in one group, and its complement to be in the other.

APPLICATION TO SEISMIC DATA

To apply this segmentation method to seismic data, the weight calculation needs to be modified. Rather than looking for clusters of pixels with similar intensity, we are now looking for groups of pixels on each side of the bright amplitude salt boundary. Therefore, we want the weights connecting pixels on either side of the salt boundary to be low and the weights connecting pixels on the same side of the salt boundary to be relatively high. Taking the negative of the maximum amplitude along the shortest path between two nodes as the weight would insure that the weights connecting pixels on either side of the salt boundary will be low. However weights on the same side would be alternating from low to high as they go from peak to trough on the seismic data. This could make the grouping more uncertain. To correct this problem, we use the maximum of the absolute value of the complex trace (instantaneous amplitude) to determine the weight.

Simply taking the minimum of the negative of the instantaneous amplitude along the shortest distance between pixels would insure that the weight would be weak at the salt boundary, however weights connecting pixels that occur within the salt boundary itself would have all be weak. This would cause the normalized cut defined in equation (1) to be very low at the pixels within the boundary and therefore the segmentation algorithm would tend cluster pixels within the salt boundary into one group. Therefore, we make weights weak only if the negative of the instantaneous amplitude along the shortest path between the nodes is less a threshold value and also a minimum value. Figure 1 shows several pixels and weights near the salt boundary.

The top of Figure 2 shows the negative of the amplitude of the complex trace of a synthetic salt boundary. This is the input for the weight calculation. The resulting weight matrix is extremely large: $(\mathtt{m} \times \mathtt{n})^2$ where $\mathtt{m}$ and $\mathtt{n}$ are the dimensions of the image. We can however view all of the weights for particular pairs of pixels. For instance, in the bottom of Figure 2 is the weight between each pixel and the adjoining pixel directly above it. Notice how it follows the peak of the amplitude.

We found that we get the best results when the maximum search distance is greater than 30 pixels. The maximum search distance is the maximum distance between pixels, beyond this distance the weights are set to zero. Using such a large search distance causes

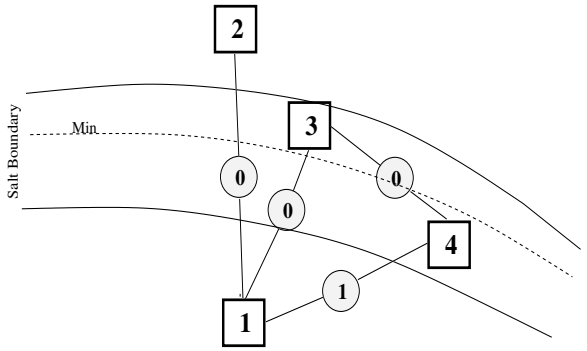

Figure 1: A cartoon illustrating some sample weights on the negative instantaneous amplitude. Several pixels are shown in squares and their corresponding weights in circles. The salt boundary is defined as areas where the negative instantaneous amplitude is below a threshold.

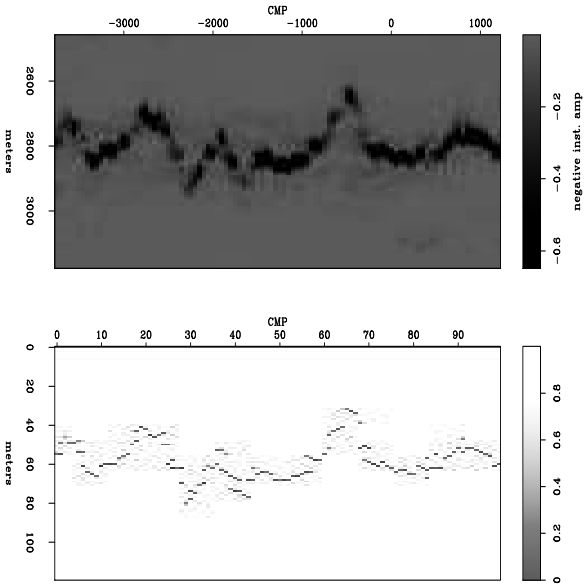

Figure 2: Top is negative absolute value of the complex trace of a synthetic salt boundary. Bottom is the weight connecting each pixel with the pixel directly above it.

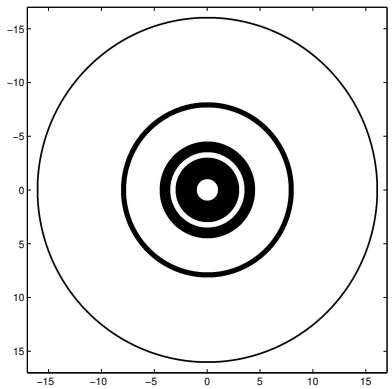

Figure 3: A cartoon of the $\log_2$ sampling of one pixel up to 16 samples.

two problems. Firstly, the matrix becomes more dense and requires more storage space. Secondly, the weights of the pixels at greater distances far outnumber those at close distances causing a bias in the weight matrix. Shi and Malik (2000) used a decaying distance weight to reduce this bias. We found that if we only use windows that are centered at powers of 2, then our matrix is still sufficiently sparse while still benefiting from greater search distances. However, this obviously causes many distances to be not sampled at all. Figure 3 shows the $\log_2$ sampling of one pixel up to a distance of 16 samples. To correct this, we plan to randomly sample from within the maximum search distance so that the number of non-zero points is approximately the same for all distances.

Iterative Velocity Analysis

An accurate velocity model is needed to image beneath a salt body. This velocity model is typically created by manually picking the top of the salt. Improvements are made to the velocity model and the data is remigrated causing the salt boundary to move and refocus. This process is repeated several times and the manual picking of the boundary can be time consuming.

Using image segmentation to pick the boundary could make this process easier. The interpreter may only be required to check the results and make some minor modifications.

Additionally, if the salt boundary changes only slightly then the eigenvector solution to equation (2) from the previous segmentation can be used as an initial solution to the current segmentation. Preliminary tests indicate that this may significantly reduce cost.

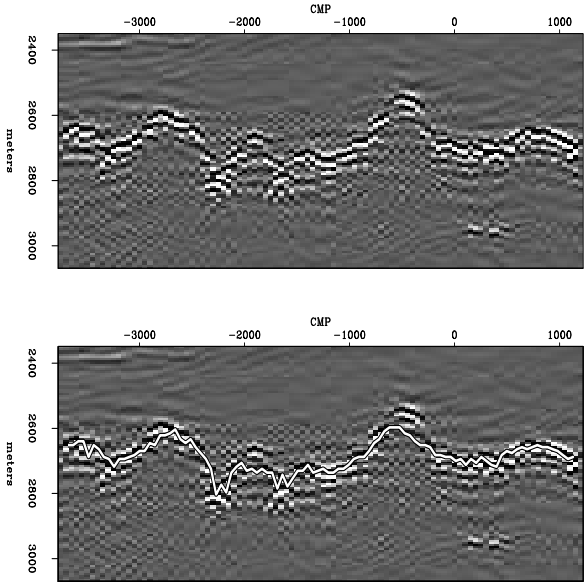

Figure 4: Top is a synthetic salt boundary. Bottom is the resulting partition from the segmentation method.

The top of Figure 4 is a synthetic 2D section that has been migrated with a preliminary velocity model. The results of applying the segmentation method can be seen in the bottom. In general it does a good job picking the salt boundary, however at cmp locations `-750` and `-1950` it has some difficulty.

The top of Figure 5 contains the eigenvector that was partitioned to get the result in Figure 4. Below is a contour plot of this eigenvector. Notice that the contours are spreading in the areas where the picking had some difficulty.

The top of Figure 6 is the same synthetic data used in Figure 4

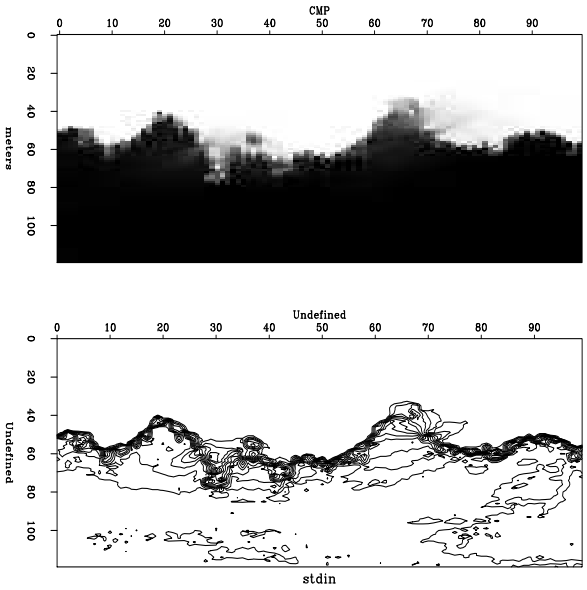

Figure 5: Top is the eigenvector used to find the boundary in the lower part of Figure 4. Bottom is a contour plot of the eigenvector. Notice the areas of uncertainty where the contours are spreading.

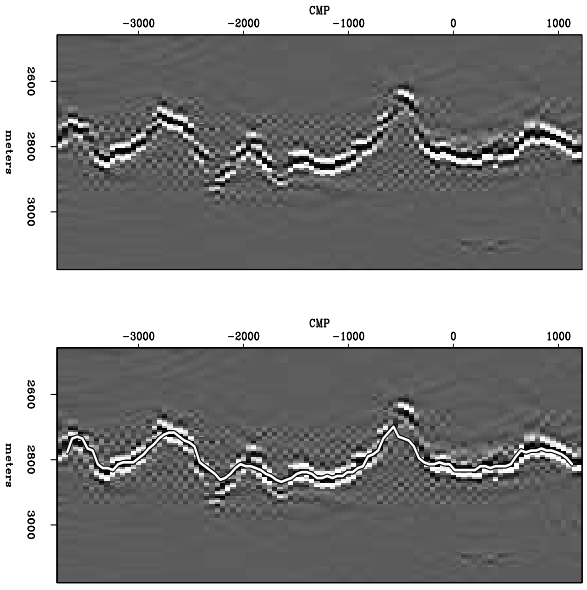

Figure 6: Top is a synthetic salt boundary after the velocity has been adjusted and remigrated. Bottom is the resulting partition from the segmentation method.

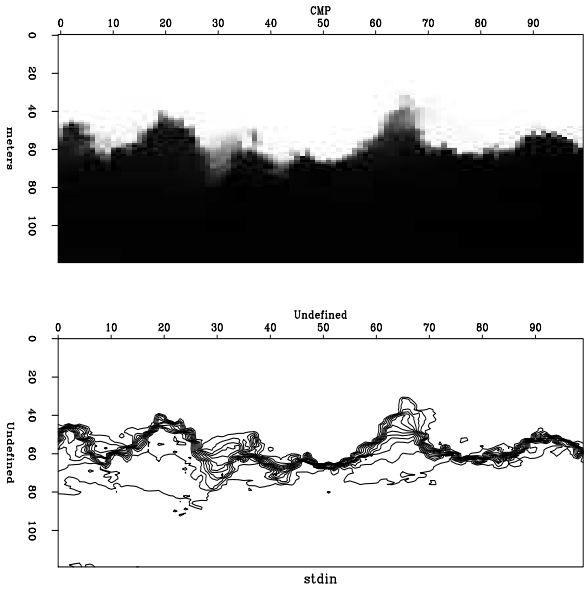

Figure 7: Top is the eigenvector used to find the boundary in the lower part of Figure 6. Bottom is a contour plot of the eigenvector. Notice the areas of uncertainty where the contours are spreading.

except migrated with an updated velocity. Again, notice that the partitioning result in the bottom successfully picks the salt boundary except in a couple of places. The corresponding eigenvector can be found in Figure 7. Notice its similarity to the eigenvector from Figure 5. Initializing subsequent segmentation processes with previous eigenvectors should speed convergence.

Multi-scale

The size of the weight matrix in equation (2) can be very large, $(\mathtt{m} \times \mathtt{n})^2$ where $\mathtt{m}$ and $\mathtt{n}$ are the dimensions of the image. Even using sparse matrices in 2 dimensions, this matrix can be prohibitively large. In 3D, things get much worse as the matrix is now $(\mathtt{m} \times \mathtt{n} \times o)^2$, where $\mathtt{o}$ is the 3rd dimension.

The segmentation method applied to a coarsely sub-sampled input cube will still select the salt boundary as long as it is the brightest amplitude in the cube. Smaller, more finely sampled cubes can then be segmented along the boundary.

CONCLUSIONS AND FUTURE WORK

Our modified segmentation method tracked the salt boundaries in our test cases. These test cases presented challenges that include non-continuous salt boundaries. An interpreter would still have to slightly adjust the tracking result in a few places.

Application of this method to 3D datasets is going to be a computational challenge that will require us to take advantage of maximizing the sparseness of the weight matrix, using accurate starting solutions, and taking a multi-scale approach.

ACKNOWLEDGMENTS

We would like to thank Frederick Billette and BP for making available to SEP the synthetic data used for our tests. We would also like to thanks Antoine Guitton and Paul Sava for suggestions and data.

REFERENCES

Hale, D., and Emanuel, J. U., 2002, Atomic meshing of seismic images: 72nd Ann. Internat. Mtg., Soc. of Expl. Geophys., Expanded Abstracts, 2126–2129.

Hale, D., and Emanuel, J. U. Seismic interpretation using global image segmentation:. Submitted for presentation at the 73rd Annual Int'l SEG Meeting, 2003. http://www.lgc.com/resources/hale03seismicinterpretationusingglobalimagesegment.pdf.

Leggett, M., Sandham, W. A., and Durrani, T. S., 1996, 3d horizon tracking using artificial neural networks: First Break, **14**, no. 11, 413–418.

Shi, J., and Malik, J., 2000, Normalized cuts and image segmentation: IEEE Trans on Pattern Analysis and Machine Intelligence, **22**, no. 8, 838–905.