Speeding up seismic image segmentation

Adam D. Halpert, Robert G. Clapp, and Biondo Biondi, Stanford University*

SUMMARY

Seismic images of the subsurface are often very large and tedious to interpret manually; as such, automatic segmentation algorithms can be highly useful for tasks such as locating large, irregularly shaped salt bodies within the images. However, seismic images present unique challenges for image segmentation algorithms. Here, a new graph-based segmentation algorithm using a "pairwise region comparison" strategy is implemented and tested on seismic images. Numerous modifications to the original algorithm are necessary to make it appropriate for use with seismic data, including changes to the nature of the input data, the way in which the graph is constructed, and the formula for calculating edge weights. Initial results indicate that the new method compares very favorably with an existing implementation of the eigenvector-based normalized cuts approach, both in terms of accuracy and efficiency. With further improvement, this method could become a useful and powerful tool for processing and interpreting seismic images.

BACKGROUND

A proper salt interpretation is a necessary component of any imaging project where salt bodies play a prominent role in the subsurface geology. Because of the sharp velocity contrast between salt and nearly any other material, inaccurate placement of salt boundaries has a disproportionate effect on the accuracy of the resulting velocity model. Such errors can have damaging imaging and engineering consequences. Unfortunately, interpreting salt boundaries is not only crucial, but also extremely tedious and time-consuming when undertaken manually. Thus, while some degree of automation would be ideal for salt picking, any such method must be highly accurate as well as efficient.

One approach to implementing automatic salt-picking is to use graph-based image segmentation. In this method, each pixel in a seismic image is treated as a node or vertex in a graph; then edges are constructed between specific pixels and weighted according to some property. Image segments are created by partitioning the graph (in this case, a partition is a salt boundary).

Graph-partitioning methods have been applied to seismic images in the past. More specifically, the eigenvector-based Normalized Cuts Image Segmentation (NCIS) algorithm (Shi and Malik, 2000) has attracted a great deal of interest because of its capability to capture global aspects of the image, rather than just track local features. Local feature trackers are ubiquitous in seismic interpretation software (for example, horizon trackers), but often struggle when they encounter situations such as a discontinuous or amplitude-varying boundary. Recent research has involved implementing the NCIS algorithm for the purpose of tracking these salt boundaries (e.g., Lomask et al. (2007), Halpert et al. (2009)). Results of this line of research are encouraging; however, there are significant limitations, most notably computational. The NCIS algorithm calls for an edge weight matrix of size n^2, where n is the number of pixels in the image; this matrix quickly grows very large, especially for 3D surveys. Calculation of eigenvectors for such a large matrix is an extremely computationally demanding task. As such, this method is limited to relatively small images; alternatively, we can restrict the computational domain so that it is limited to a specific region around a previously interpreted boundary. However, this means the method is of little use if there is no "best guess" model available, or if the accuracy of that model is in question.

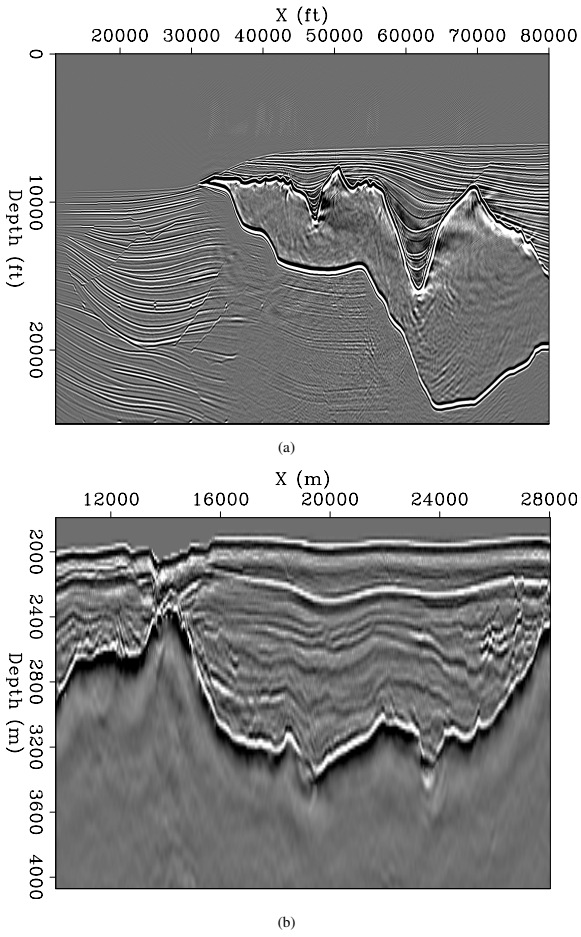


Figure 1: A perfect-velocity migration of the Sigsbee synthetic model (a), and a field seismic image (b) that will be used as examples.

Thus, a more efficient global segmentation scheme that can include the entire image in the computational domain would be

a very useful tool for interpretation of seismic images. One candidate for such a scheme is the algorithm from Felzenszwalb and Huttenlocher (2004), who write that their algorithm is both "highly efficient" and also "captures non-local properties" when segmenting images. These two features are crucial for the task of seismic image segmentation. The algorithm is designed to run in $O(n \log n)$ time, where n is the number of pixels in the graph; in contrast, other methods such as NCIS require closer to $O(n^2)$ time to run. This represents a significant cost savings, especially for very large 3D seismic datasets that are becoming increasingly common. The new algorithm relies heavily on the concept of the "Minimum Spanning Tree" (see Zahn (1971)), which is a non-circuitous connected graph requiring the minimum sum of edge weights. The MST concept allows Felzenszwalb and Huttenlocher (2004) to develop what they term a "pairwise region comparison" predicate in order to determine whether two regions should be considered separate segments of the graph, or merged into a single region. While this procedure is relatively simple, it is designed to allow highly heterogeneous regions to be segmented as a single component of an image. Additionally, the authors note that their algorithm produces segmentations that are "neither too coarse nor too fine," referring to the global capabilities of the segmentation process.

APPROACH

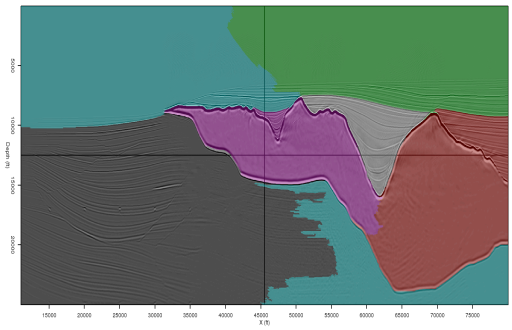
The goal of this paper is to apply the algorithm of Felzenszwalb and Huttenlocher (2004), introduced above, to seismic data (specifically, the example images in Figure 1). Publicly-available code from Felzenszwalb (2010) allows for relatively easy implementation for standard images. However, seismic images are very different from photographs or other types of images. The following sections describe the rationale and procedure for modifying and adding additional features to the algorithm in order to apply it to seismic data.

Performance of unaltered algorithm

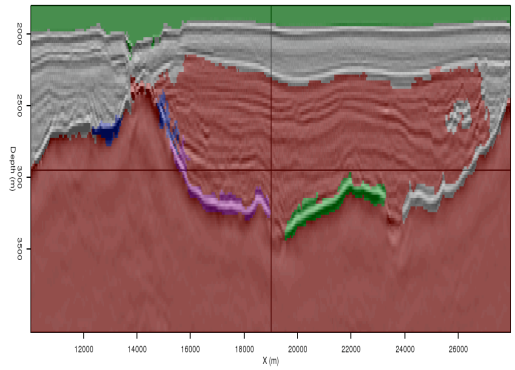
Once the algorithm is modified to accept seismic data, seismic images may be segmented according to the same rules used by Felzenszwalb and Huttenlocher (2004) to segment RGB images. The results for the examples introduced earlier can be seen in Figures 2(a) and 2(b). In these figures, each segment or region is assigned a random color, and the segments are overlain on the seismic image itself for reference. The results from the (relatively) unaltered algorithm are promising, but are clearly lacking. The field data result in Figure 2(b) is especially poor.

Transformation of input data

Seismic data are a function of amplitude and phase, presenting a challenge for any segmentation algorithm. We may see an indication of this in Figure 2(b), where the algorithm interprets the area around the boundary as several regions, instead of an interface between just two regions. In this case, the boundary itself becomes its own "region" in several locations. To avoid this situation, we simply calculate the "amplitude of the envelope" of the data in order to ensure that amplitude information,



(a)



(b)

Figure 2: Segmentation of the example seismic images from Figure 1, using the original algorithm from Felzenszwalb and Huttenlocher (2004).

and not phase, is being used to segment the image. In this way, it is much more likely that the unwanted behavior seen in the original examples can be mitigated.

Creating the graph

The original implementation of the pairwise region comparison algorithm from Felzenszwalb (2010) creates a graph with eight edges per node (pixel). This graph is constructed by looping over every pixel, and performing four calculations at each vertex. The left side of Figure 3 attempts to illustrate this process – if the "active" pixel is the one in red, edges are built to each of the blue pixels. Since every pixel in the image undergoes this process, a form of reciprocity allows for each pixel to be connected to its eight immediate neighbors via edges. While this process allows for the extreme efficiency of the algorithm, the unique and often irregular nature of seismic data does not lend itself well to segmentations using so few edges per vertex or pixel. Instead, a much larger "stencil," seen on the right of Figure 3, has been implemented. This allows

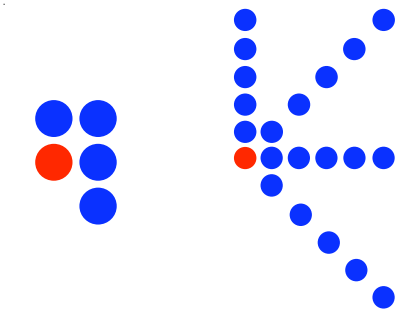


Figure 3: Stencils used for comparing pixel values and assigning edge weights for the graph. At left, the five-point stencil (8 edges per pixel) used in the original implementation from Felzenszwalb and Huttenlocher (2004); at right, a modified 21-point stencil (40 edges per pixel) used for the seismic images.

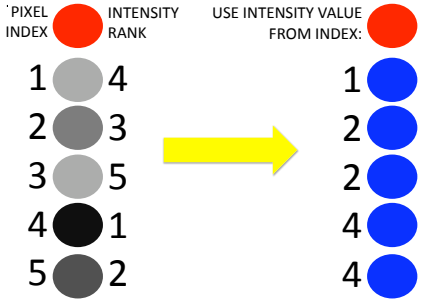


Figure 4: Diagram illustrating the logic behind deciding which pixel intensity value to use when calculating edge weights. Pixel intensities are shown and ranked on the left; the numbers in the right column indicate which intensity value will be used when calculating the edge weight between the adjacent blue pixel and the pixel in red.

for many more comparisons (40) per pixel, and a far greater amount of information goes into the segmentation algorithm. While this approach obviously decreases the efficiency of the algorithm, the increased accuracy seen in the final results appears to make it a worthwhile trade-off. Even with the sharply increased number of edges per node, this algorithm is still far less computationally intensive than the NCIS algorithm from Shi and Malik (2000).

Calculating edge weights

Unlike image types for which the algorithm was originally designed, seismic images have regions defined by their *boundaries* rather than the character of their interiors. Therefore, we must create a process for calculating edge weights that treats a boundary *between* two vertices as more convincing evidence for the existence of two regions than simply a difference in intensity at the two pixels themselves. The stencil used for determining graph edges forms what are essentially four line "segments" at each vertex – one horizontal, one vertical, and two diagonal. For each segment, we use the largest intensity value of any pixel between the two endpoint vertices for the purposes of determinining likelihood of a boundary. Figure 4 illustrates the logic behind this process.

Once we have selected the intensity value to use for determining the edge weight, we must still calculate the weight value itself. Rather than using a simple intensity difference calculation between the two pixels (used in the implementation from Felzenszwalb (2010)), we have found it more appropriate to use an exponential function. Additionally, since the edges in the graph can now be much longer than with the adjacent-pixels-only approach taken in the original implementation, it makes sense to include a distance-weighting term to the edge weight calculation:

$$w_{ij} = \exp((\max I(\mathbf{p}_{ij}))^2)\exp(d_{ij}), \qquad (1)$$

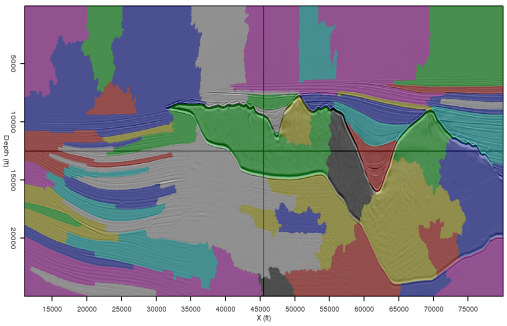
where $\mathbf{p}_{ij}$ is the vector of all pixels between i and j and d_{ij} is simply the Euclidean distance (in samples) between the two pixels.

Once each of the edges is assigned a weight, the segmentation of the image can proceed as described in Felzenszwalb and Huttenlocher (2004). In summary, the process begins with each pixel as its own image segment; then individual pixels, and eventually, groups of pixels, are merged according to thresholding criteria. Segments can also be merged in post-processing if they are smaller than a "minimum segment size" parameter specified by the user.

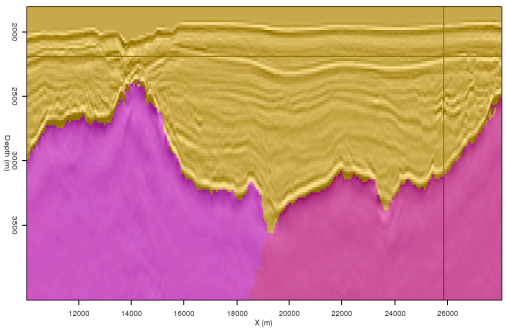
RESULTS

The results of the procedure detailed in the sections above can be seen in Figure 5. In Figure 5(a), although the salt body has been segmented into several regions, all of these regions are contained within the salt body and fully conform to its boundaries. This represents an improvement over the original segmentation in Figure 2(a). The improvement is even more dramatic for the field data example (original image in Figure 1(b)). In Figure 5(b), we see that the salt body has been segmented virtually perfectly; no longer do the salt body segments spill over the boundary, nor are parts of the boundary itself treated as individual segments.

A major motivation behind this research was to attempt to improve on segmentation results using the NCIS method from Shi and Malik (2000), and adapted for use with seismic images by Lomask et al. (2007). The first indication that this method does indeed represent a substantial improvement is the fact that the synthetic image from Figure 1(a) is simply *too large* to be segmented on a single processor using the existing NCIS implementation. The necessity of holding a giant, sparse weight matrix in memory and calculating an eigenvector precludes problems of this size from being feasible. The smaller field data example, however, is well-suited for the NCIS algorithm, and will allow us to make a relatively fair comparison. Figure 6 shows the boudaries calculated by the two segmentation methods: the NCIS result in green, and the new PRC



(a)



(b)

Figure 5: Results of applying the modified segmentation algorithm to the example images from Figure 1. Both (a) and (b) are a significant improvement over the original results in Figure 2.

result in pink. The two results are nearly identical, serving to increase confidence in the new method.

One of the primary means of comparison for the relative effectiveness of these two approaches to image segmentation is the computational efficiency of the method. Table 1 summarizes the computational expense required to create the examples seen in this paper. Again, due to memory constraints the existing NCIS implementation is unable to segment an image the size of Figure 1(a). The implementation described here, however, produces an accurate segmentation in 31 seconds; during this time, approximately 55 million edges are created, weighted, and used to segment the graph. The efficiency advantage for the new implementation is quantified using the field data example; in this case, the image is segmented over 150 times faster using the new method. These differences are extremely significant and represent a huge savings of time and computational expense, especially for larger problems.

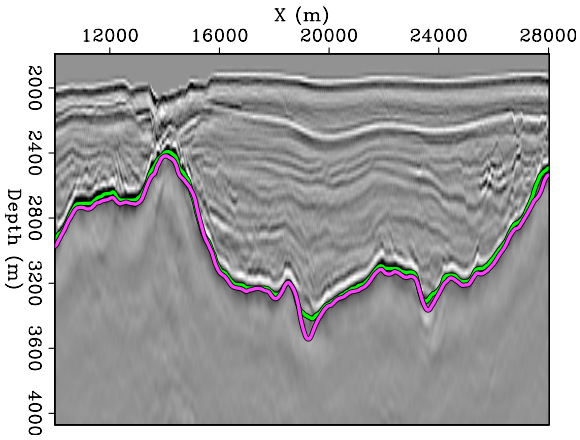


Figure 6: Comparison of the boundaries obtained using the NCIS eigenvector method (green) and the newer pairwise region comparison method (pink).

Table 1: Comparison of CPU times for the two methods

		CPU time (s)	
Image type	**Pixels**	NCIS	PRC
Synthetic data	2761000	n/a	31
Field data	55000	156	1

CONCLUSIONS

Initial results from applying the modified Pairwise Region Comparison (PRC) algorithm to both synthetic and field seismic images are extremely encouraging. Segmentation of the synthetic image (Figure 5(a)) accurately locates the boundaries of the salt body, although several different segments are required. Segmentation of the field seismic data (Figure 5(b)) was even more successful. Compared to an existing implementation of the Normalized Cuts algorithm from Shi and Malik (2000), the new method performs extremely well – it required only half a minute to segment the synthetic data image (which is too large for the NCIS implementation to handle), and only one second to segment the smaller field data example, over 150 times faster than NCIS. An additional advantage is that the newer algorithm is able to operate on the entire image, rather than only within a certain windowed radius of a previously interpreted boundary. This has many advantages, not least of which is the opportunity to identify segments other than only salt bodies. Instead of a binary salt/no-salt determination, the ability to identify coherent sedimentary "segments" as well would be tremendously useful for constructing seismic velocity models.

ACKNOWLEDGMENTS

We thank SMAART JV and Unocal (now Chevron) for providing the data used for examples, and the sponsors of the Stanford Exploration Project for their support.

EDITED REFERENCES

Note: This reference list is a copy-edited version of the reference list submitted by the author. Reference lists for the 2010 SEG Technical Program Expanded Abstracts have been copy edited so that references provided with the online metadata for each paper will achieve a high degree of linking to cited sources that appear on the Web.

REFERENCES

Felzenszwalb, P. F., 2010, Image segmentation: (http://people.cs.uchicago.edu/~pff/segment/).

Felzenszwalb, P. F., and D. P. Huttenlocher, 2004, Efficient graph-based image segmentation: International Journal of Computer Vision, **59**, no. 2, 167–181, doi:10.1023/B:VISI.0000022288.19776.77.

Halpert, A., R. G. Clapp, and B. L. Biondi, 2009, Seismic image segmentation with multiple attributes: 79th Annual International Meeting, SEG, Expanded Abstracts, 3700–3704.

Lomask, J., R. G. Clapp, and B. Biondi, 2007, Application of image segmentation to tracking 3D salt boundaries: Geophysics, **72**, no. 4, P47–P56, doi:10.1190/1.2732553.

Shi, J., and J. Malik, 2000, Normalized cuts and image segmentation: IEEE Transactions on Pattern Analysis and Machine Intelligence, **22**, 838–905.

Zahn, C. T., 1971, Graph-theoretical methods for detecting and describing gestalt clusters: IEEE Transactions on Computers, **C-20**, no. 1, 68–86, doi:10.1109/T-C.1971.223083.