Salt delineation via interpreter-guided 3D seismic image segmentation

Adam Halpert

ABSTRACT

Although it is a crucial component of seismic velocity model building, salt delineation is often a major bottleneck in the interpretation workflow. Automatic methods like image segmentation can help alleviate this bottleneck, but issues with accuracy and efficiency can hinder their effectiveness. However, a new graph-based segmentation algorithm can, after modifications to account for the unique nature of seismic data, quickly and accurately delineate salt bodies on 3D seismic images. In areas where salt boundaries are poorly imaged, limited manual interpretations can be used to guide the automatic segmentation, allowing for interpreter insight to be combined with modern computational capabilities. A successful 3D field data example demonstrates that this method could become an important tool for interactive interpretations tasks.

INTRODUCTION

Salt interpretation is a vital component of seismic imaging projects in many of the world's resource-rich basins. The sharp contrast between seismic velocities within salt structures and those in the surrounding sediments means that inaccurate interpretation of these salt-sediment boundaries can lead to severe degradation of images sub-salt; this is of particular concern since sub-salt reservoirs are often the targets for modern exploration. Unfortunately, salt interpretation is not only critical, but often time-consuming and human-intensive as well. For large 3D surveys, manual salt-picking can consume significant resources during model-building workflows that stretch for weeks or months. This can be exacerbated by iterative sediment- and salt-flooding techniques that require several rounds of salt interpretation (Mosher et al., 2007). The semi-automatic image segmentation method we present here aims to help alleviate this bottleneck, while maintaining the accuracy necessary for successful model building and imaging.

While image segmentation is most often associated with fields such as medical imaging and photo processing, several efforts have been made to apply automatic segmentation concepts to seismic images. A variety of approaches has been tried, including pixel-by-pixel classifier methods using fuzzy math (Valet et al., 2001) or texture attributes (Berthelot et al., 2012). These methods can incorporate interpreter input by "training" the algorithm through the use of if-then guidelines or training images. Another category of methods that has proven popular for seismic images is known as 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 (for example, a partition may represent a salt boundary). An advantage of graph-based segmentation is that it provides a globally optimum solution to the segmentation problem. This compares favorably with automatic interpretation tools such as horizon trackers that tend to get ”lost” if a boundary becomes chaotic or discontinuous.

The first graph partitioning seismic image segmentation algorithms were adapted from the eigenvector-based Normalized Cuts Image Segmentation (NCIS) method (Shi and Malik, 2000). One of the first applications was for atomic meshing of seismic images (Hale and Emanuel, 2002, 2003), followed by efforts to track salt boundaries (Lomask et al., 2007; Lomask, 2007). The method was effective, but faced limitations - most notably computational. The NCIS algorithm requires the calculation of eigenvectors 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. Despite modifications to limit the computational domain of this method, it remains infeasible for very large 3D images. The method we present relies instead on the graph-based technique of Felzenszwalb and Huttenlocher (2004), which was designed with efficiency as a primary consideration. With modifications to account for the unique properties of seismic images, this method can accurately segment 3D images at a fraction of the expense required for the eigenvector approach. Furthermore, valuable interpreter insight can be incorporated in the form of limited 2D interpretations, which are then used to guide an automatic 3D segmentation.

SEGMENTATION METHOD

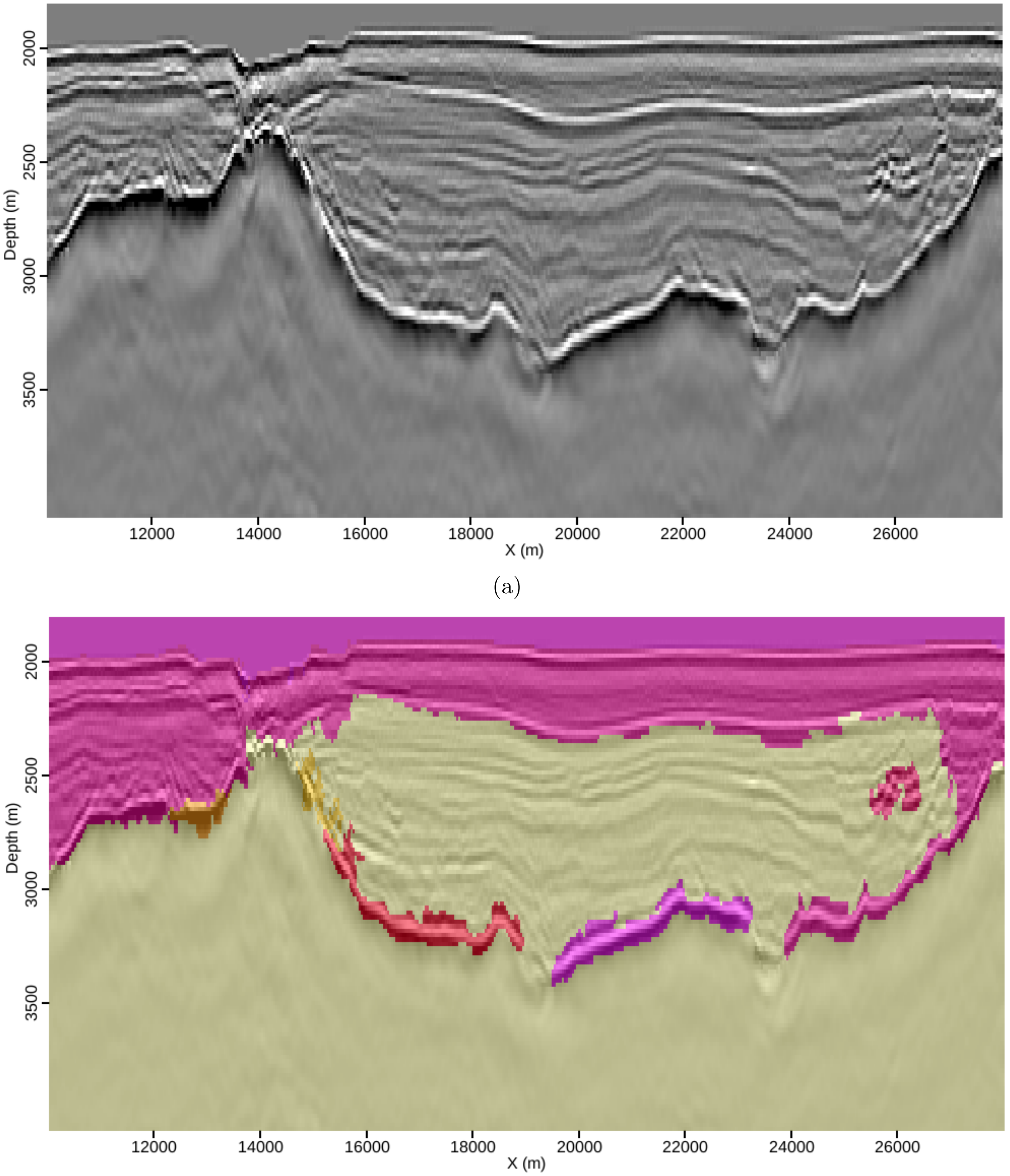
The algorithm developed by Felzenszwalb and Huttenlocher (2004) is designed such that its computational requirements scale at approximately $n \log n$ (where n is the number of pixels in the image), a significant cost savings over other graph-based approaches scaling at n^2. The algorithm relies heavily on the concept of the “Minimum Spanning Tree” (Zahn, 1971) of a graph. A graph’s edges are weighted using a measure of dissimilarity between vertex pairs; a connected graph is defined as one in which all such edges are assigned a weight value. If a spanning tree is a connected graph which connects all vertices of the graph without forming a circuit, the minimum spanning tree (MST) of a graph is the spanning tree requiring the minimum sum of edge weights. By sorting an image’s edge weights in increasing order, the MST concept allows Felzenszwalb and Huttenlocher (2004) to develop what they term a “pairwise region comparison” (PRC) predicate in order to determine whether two regions should be considered separate segments of the graph, or merged into a single region. Briefly, the method examines the relationship between edges connecting pixels within a region, and edges connecting pixels across a putative boundary. If these two groups are similar to within a threshold, the two regions are merged. This process allows even highly heterogeneous regions to be segmented as a single component of an image – an important capability when handling noisy images.

Adaptation for seismic images

Seismic images are distinct in many ways from more conventional photographs and medical images for which this and most image segmentation algorithms are designed. The effects of this fact can be seen in Figure 1(b), the result of using the un-altered PRC algorithm to segment a 2D image from the Gulf of Mexico (Figure 1(a)). In this and all subsequent depictions of segmentation results, the interpreted segments are assigned a random color and overlaid on the image for reference. An initial hurdle is that seismic data are a function of amplitude and phase, presenting a challenge for any segmentation algorithm; in Figure 1(b), the algorithm interprets the area around the salt boundary as several regions, instead of an interface between just two regions. This problem can be mitigated by using the amplitude of the seismic signal's envelope as the input for segmentation, rather than the raw image in Figure 1(a). A second concern is that regions such as salt bodies in a seismic image are most easily delineated by their boundaries, rather than, for example, color attributes used to segment photographs. Therefore, modifications to the algorithm's procedure for both constructing the graph and weighting its edges are required to obtain acceptable segmentation results for seismic images.

The original implementation of the pairwise region comparison (PRC) algorithm 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 2 illustrates 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 larger "stencil," such as the one on the right in Figure 2, has been implemented. The length of the stencil's arms is a user-defined parameter which may be adjusted based on data quality; larger stencils should be used for noiser data, but the trade-off is increased computational complexity. Increasing the size of the stencil allows for many more comparisons 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 increased number of edges per node, this algorithm is still far less computationally intensive than the NCIS algorithm from Shi and Malik (2000).

Finally, the edges constructed using the modified stencil in Figure 2 must be



(a)

(b)

Figure 1: (a) A 2D field seismic image, and (b) its corresponding segmentation using the original algorithm from Felzenszwalb and Huttenlocher (2004). [**ER**]

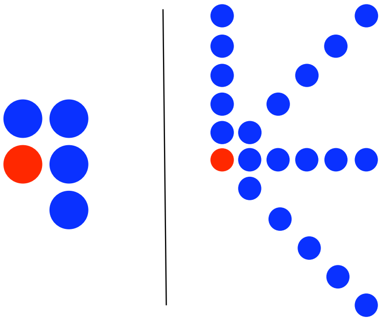


Figure 2: Stencils used for comparing pixel values and assigning edge weights for the graph. At left, the five-point stencil (8 edges per pixel) used in the original implementation from Felzenszwalb and Huttenlocher (2004); at right, a modified 21-point stencil (40 edges per pixel) used for the seismic images. For 3D images, additional stencil arms extend into the third dimension; the length of the stencil arms is a user-defined parameter. [**NR**]

weighted in a manner 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. When determining the weight for an edge with an endpoint along one arm of the stencil in Figure 2, we use the largest intensity value of any pixel between the two endpoints. For example, a high intensity value along one arm of the stencil would suggest that that particular arm intersects a boundary. Figure 2 illustrates the logic behind this process.

Once we have selected the intensity value to use for determining the edge weight, the weight value is cacluated using an exponential function:

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

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. The distance-weighting d term accounts for the fact that the edges in the graph can now be much longer than with the adjacent-pixels-only approach taken in the original implementation.

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. Figure 4 is the much-improved result when the example image in Figure 1(a) is segmented by the modified PRC algorithm.

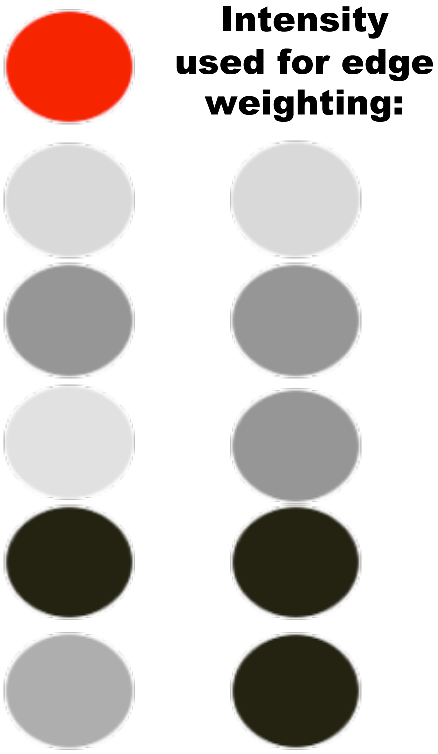


Figure 3: Diagram illustrating the logic behind deciding which pixel intensity value to use when calculating edge weights. Pixel intensities along one "arm" of the stencil in Figure 2 are shown on the left; darker colors represent higher intensities. The right column indicates which intensity value will be used when calculating the edge weight between the "active" (red) pixel and the adjacent pixel. [**NR**]

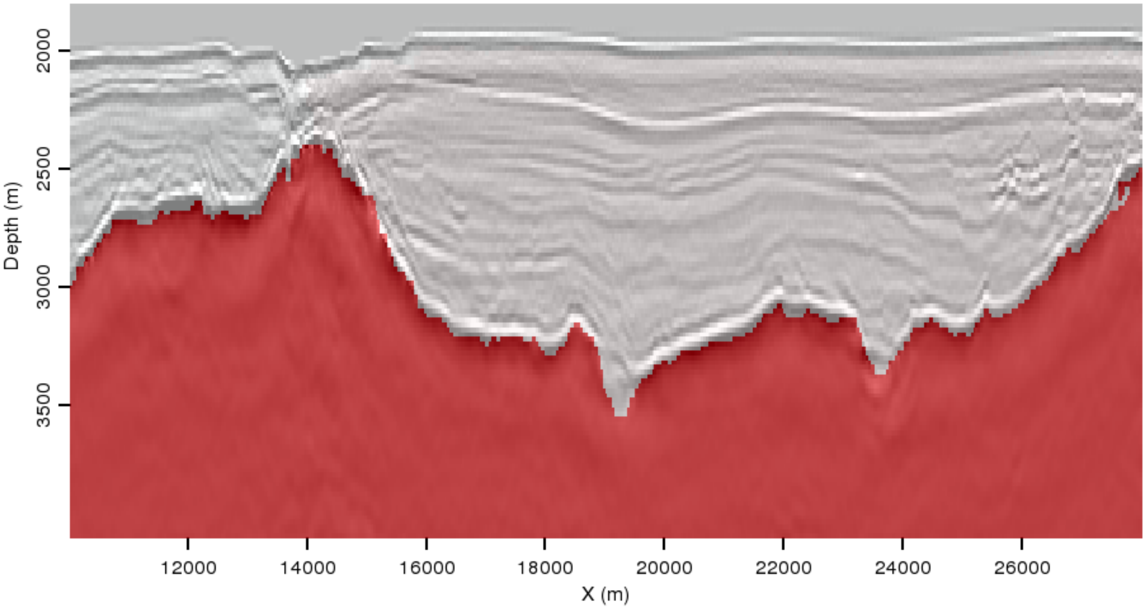


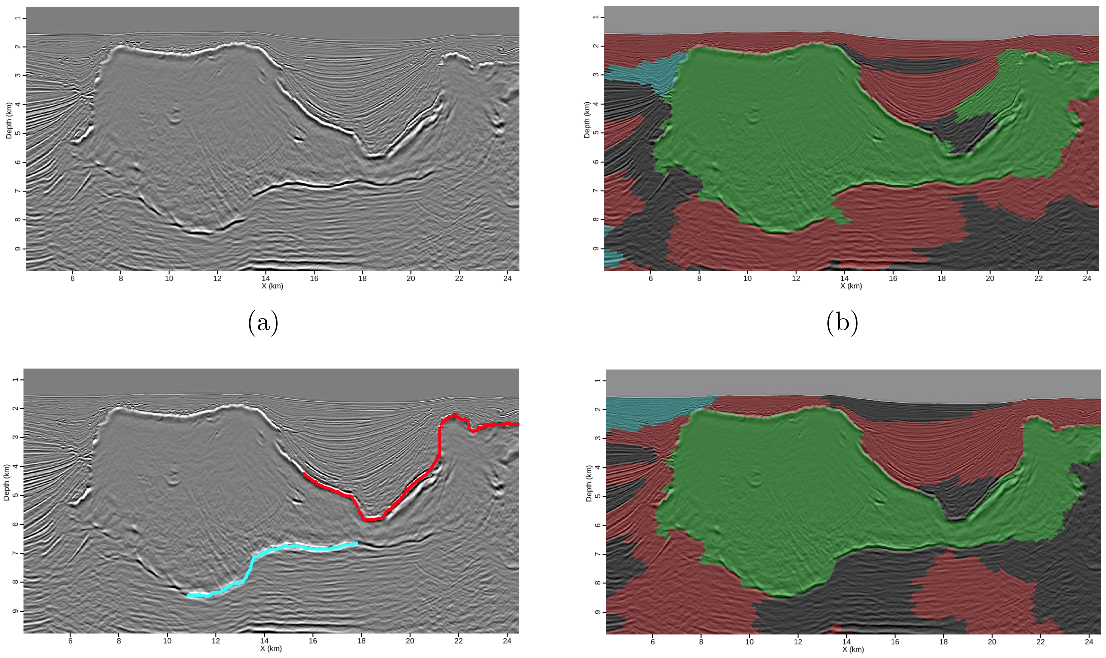
Figure 4: Final segmentation result of the image in Figure 1(a), after modifications to the algorithm. [**ER**]

INTERPRETER GUIDANCE

Unfortunately, a fully-automatic method will often be insufficient for obtaining an acceptable salt interpretation; acquisition, model-building, and imaging challenges all contribute to situations in which salt boundaries appear faint, discontinuous, or not to be present at all. In another 2D Gulf of Mexico example image (Figure 5(a)), for example, there are locations along both the top and base of the salt body where the boundary is poorly imaged. This can result in ”leakage” of the automatically interpreted salt segments (Figure 5(b)). In these cases, valuable interpreter insight should be incorporated into the procedure. Figure 5(c) shows manual salt boundary interpretations in areas where leakage is apparent in Figure 5(b). The most efficient way to include this information in the PRC algorithm is to modify the input image by increasing intensity values at the manual pick locations. Instead of assigning arbitrarily large values, however, we define a new amplitude value (A) for a “picked” pixel at position (x,y,z) in terms of the highest-amplitude pixel in a neighborhood surrounding it and a scaling factor α:

$$A_{xyz} = \alpha \max_{|x-i|\le 5, |y-j|\le 5, |z-k|\le 5} A_{ijk}. \qquad (2)$$

This ensures that the picked boundary will not appear radically different from its surroundings, which could present challenges for the automatic segmentation algorithm. Now, segmenting the new input image with parameters identical to the original segmentation yields the result seen in Figure 5(d). The segments conform to the manual picks seen in Figure 5(b), while the rest of the image is segmented as accurately as the original result in Figure 5(b).



(a)

(b)

(c)

(d)

Figure 5: (a) A 2D section from the Gulf of Mexico; (b) Segmentation result using the unaltered algorithim from Felzenszwalb and Huttenlocher (2004); (c) Manual salt picks supplied to guide the automatic segmentation; (d) Segmentation result after interpreter guidance. [**ER**]

Additional modifications are required for 3D images. Because segments are much larger in 3D, amplitude changes on a single 2D section are not significant enough to alter 3D segmentation results. Instead, we must "project" an interpreter's manual picks on an inline section, into the third (crossline) dimension. To do this, we make the assumption that a salt boundary will not fluctuate by more than two pixels per slice in the crossline direction, and construct a square pyramid in the crossline direction like the one depicted in Figure 6. The pyramid has sides of length $2h$, where h is the number of crossline samples between the base of the pyramid and its apex, which is the manually interpreted point. Now, for any pixel Q that falls within a pyramid with an apex at point P, the new amplitude value at point Q is

$$A^Q_{\text{new}} = A^Q_{\text{orig}} + \frac{A_0}{||PQ||}, \tag{3}$$

where A_0 is the amplitude value at point P as determined by equation 2, and $||PQ||$ is the distance between the two points. The expression is additive to ensure that any hint of the boundary already present will not be overwhelmed by the interpretation on a nearby slice.

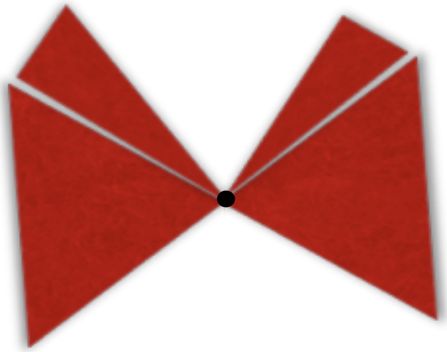
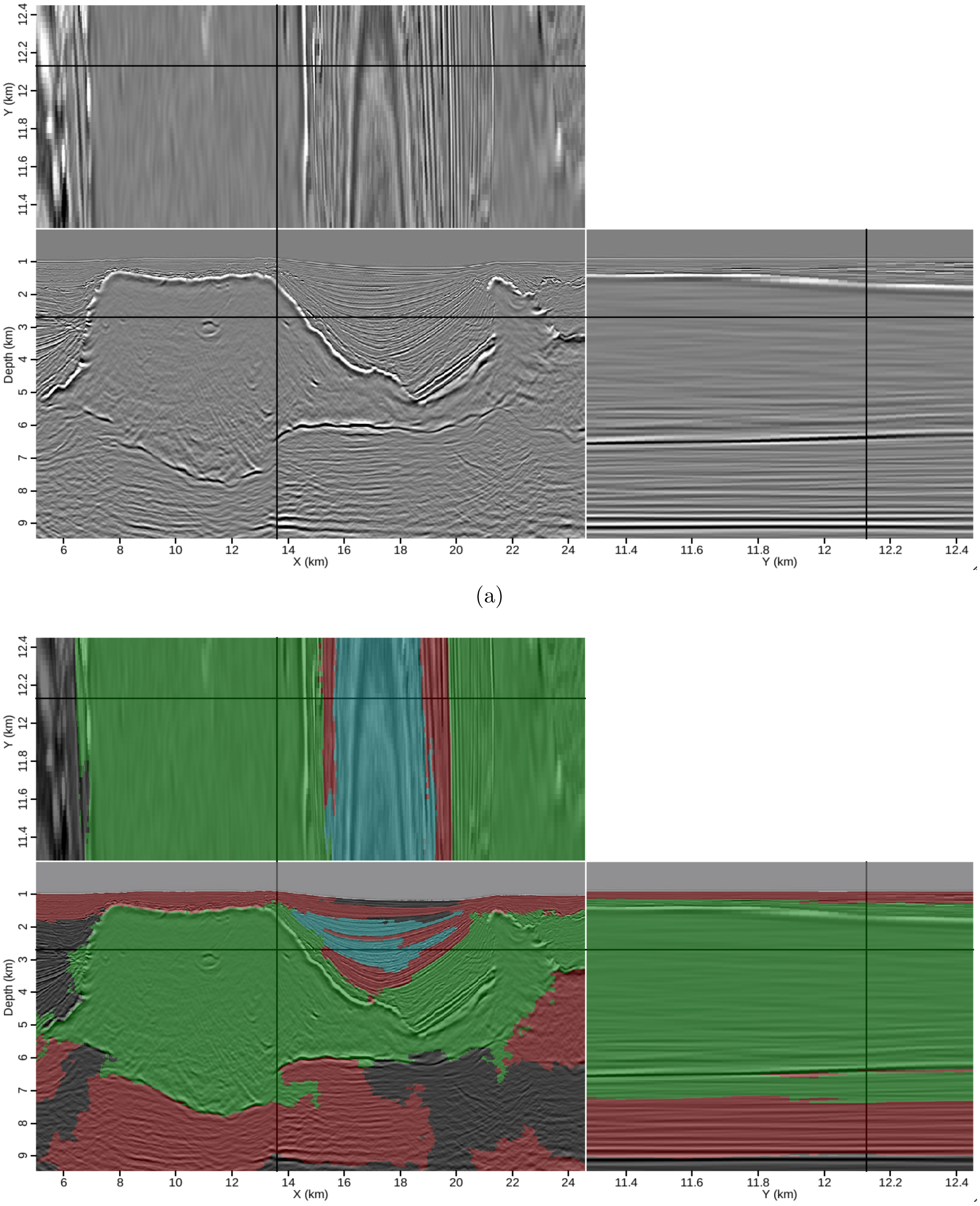


Figure 6: Depiction of the pyramid used to "project" an interpreter's picks from a single 2D slice into the third dimension. The influence of the pick (at the apex of the pyramid) decays with distance. [**NR**]

3D FIELD DATA EXAMPLE

Figure 7(a) shows slices through a 3D image cube from a Gulf of Mexico dataset provided by WesternGeco. From the initial segmentation result (Figure 7(b)), it is clear that the salt boundary discontinuities present challenges for the automatic segmentation algorithm. To correct the apparent leakages, manual interpretations are supplied for selected locations at two crossline locations (Figure 8). The effects of these manual picks on the input amplitude data are shown in Figures 9(a) and 9(b). Not only are higher amplitudes obvious at the pick locations themselves, but the procedure described in the previous section has clearly influenced the intensity values at neighboring crossline values. Now, the updated segmentation result (Figure 10) is improved on both the inline and crossline sections. To emphasize the improvement, Figure 11(a) is the initial segmentation result shown far from any of the manual pick locations, while Figure 11(b) is the result after interpreter guidance. Even far from



(a)

(b)

Figure 7: (a) Slices through a 3D image cube from the Gulf of Mexico; (b) Segmentation result prior to interpreter guidance. [**CR**]

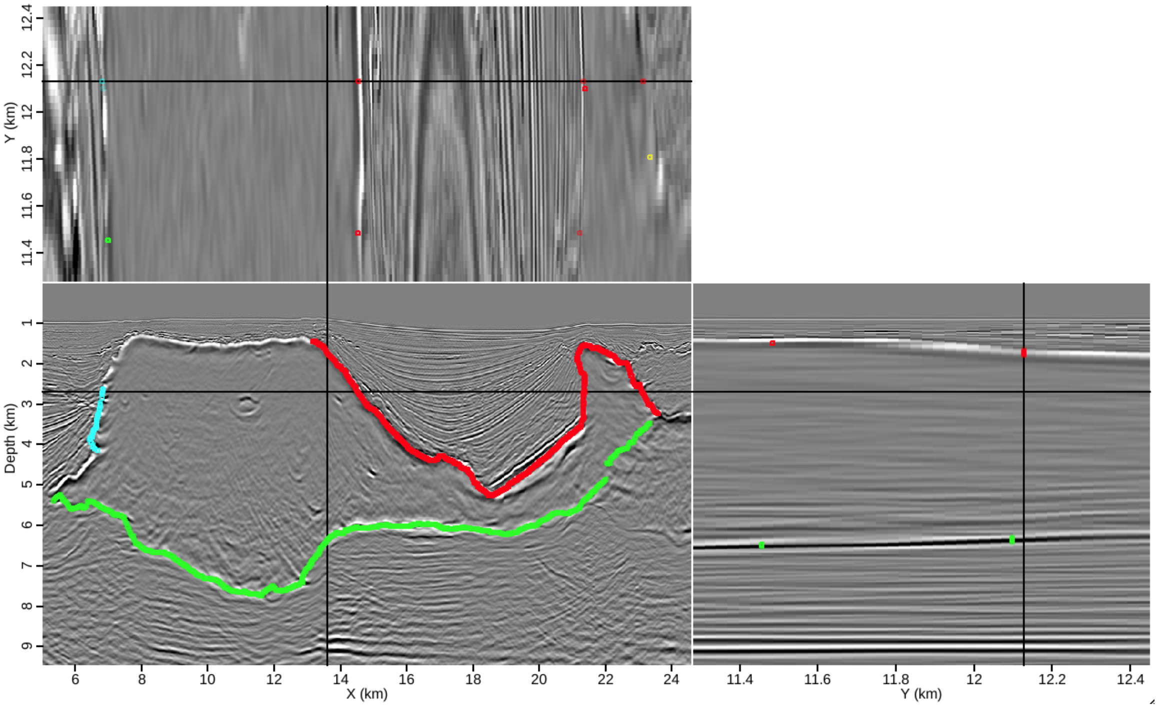


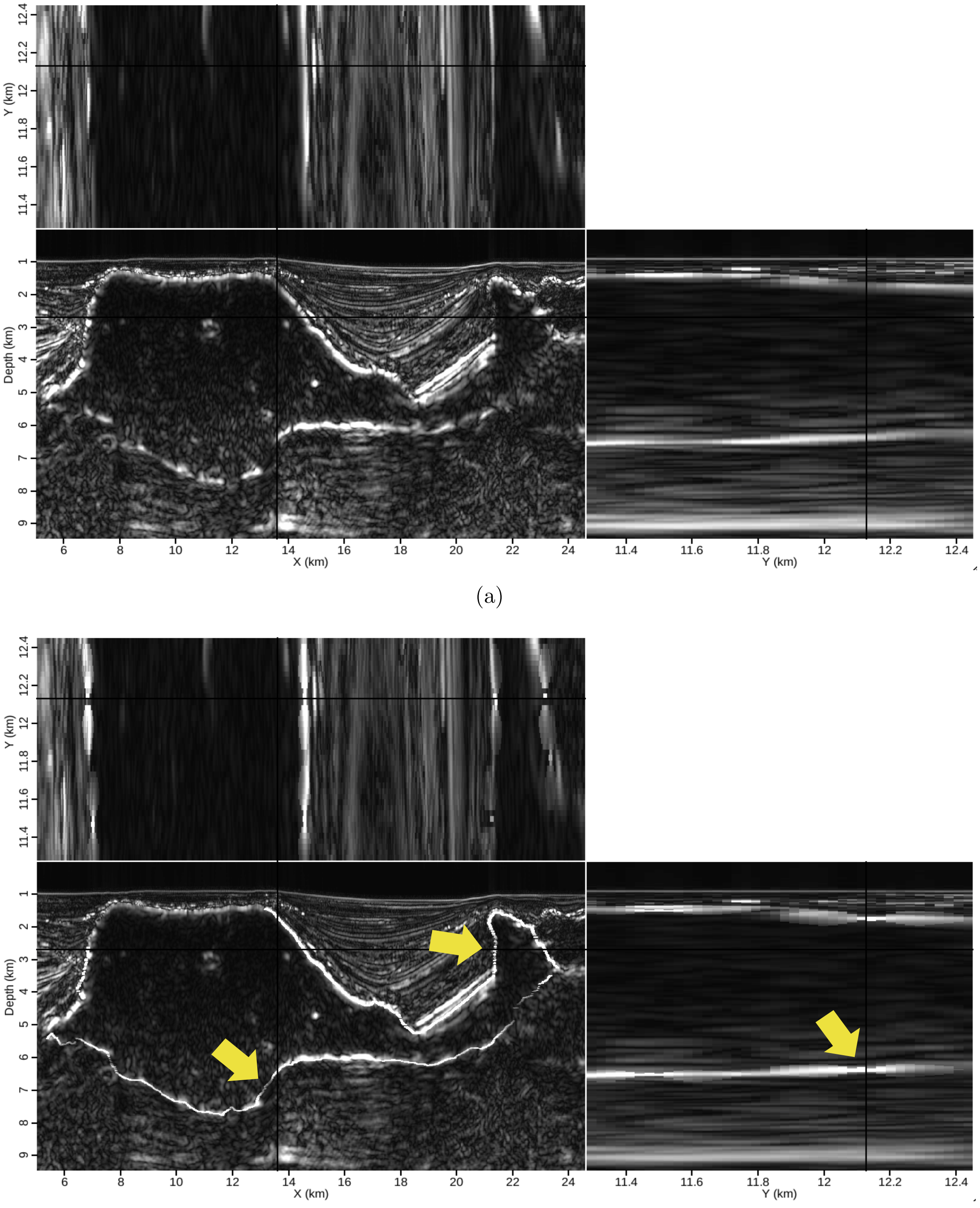
Figure 8: Manually-interpreted salt picks used to guide the automatic 3D segmentation. [**ER**]

the actual picks, the automatic segmentation process is significantly more accurate when incorporating interpreter guidance.

As expected, the algorithm does operate extremely efficiently. This 3D example had over 30 million pixels, and over 700 million graph edges were constructed. A single CPU performed the segmentation in less than three minutes, highly efficient compared to other segmentation techniques. Furthermore, this method operates on the entire image cube, rather than a limited domain around the salt body. This opens the door for additional interpretation aids within the algorithm's capabilities, such as stratigraphic segmentation.

CONCLUSIONS

Applying the modified Pairwise Region Comparison (PRC) segmentation algorithm to 2D and 3D field seismic images allows for accurate, semi-automatic salt structure delineation. While fully automatic segmentations are sometimes successful, limited manual interpretations on one or more 2D slices can be used to guide a 3D segmentation process. This allows for improved results throughout the image cube, not just near manual pick locations. The new algorithm performs extremely efficiently compared to other automatic interpretation techniques, and operates on the full seismic image or cube rather than a limited domain around a supposed salt structure. There-



(a)

(b)

Figure 9: Amplitude of the envelope volumes (a) before and (b) after modification according to the interpreter guidance scheme. Changes are particularly noticeable at the indicated locations. [**ER**]

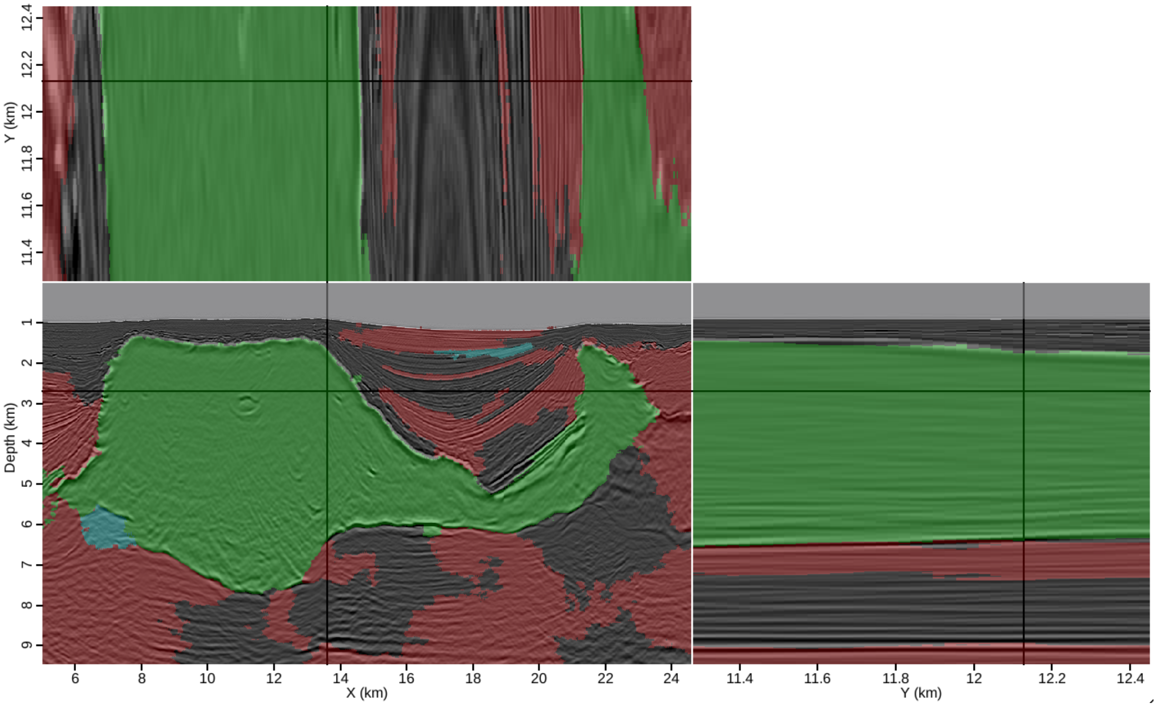


Figure 10: Segmentation result incorporating interpreter guidance. This result is much more accurate than the initial result in Figure 7(b). [**CR**]

fore, this could make it an important tool for interactive interpretation procedures that can streamline the model building workflow.

ACKNOWLEDGMENTS

I thank Unocal (now Chevron) and WesternGeco for providing the data used for examples, and all SEP sponsors for their support.

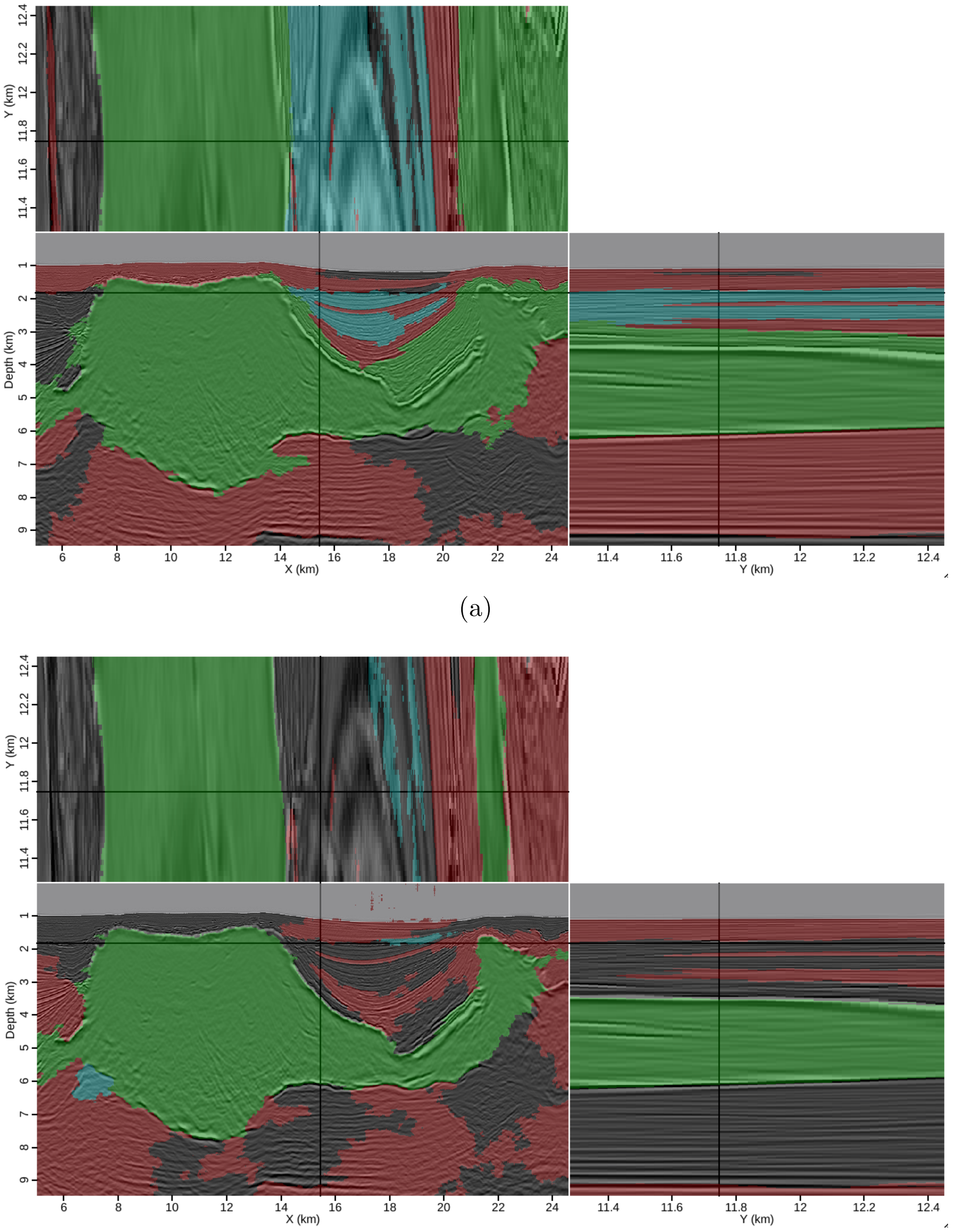
REFERENCES

Berthelot, A., A. H. S. Solberg, E. Morisbak, and L.-J. Gelius, 2012, 3[d] segmentation of salt using texture attributes: SEG Technical Program Expanded Abstracts 2012, 1–5.

Felzenszwalb, P. F. and D. P. Huttenlocher, 2004, Efficient graph-based image segmentation: International Journal of Computer Vision, **59**, 167–181.

Hale, D. and J. U. Emanuel, 2002, Atomic meshing of seismic images, *in* Expanded Abstracts, 2126–2129, SEG.

———, 2003, Seismic interpretation using global image segmentation, *in* Expanded Abstracts, 2410–2413, SEG.



(a)

(b)

Figure 11: Segmentation results for another set of slices through the 3D cube, (a) before and (b) after interpreter guidance. The result is greatly improved, even far from where the manual salt picks were supplied. [**CR**]

Lomask, J., 2007, Seismic volumetric flattening and segmentation: PhD thesis, Stanford University.

Lomask, J., R. G. Clapp, and B. Biondi, 2007, Application of image segmentation to tracking 3ṣalt boundaries: Geophysics, **72**, P47–P56.

Mosher, C., E. Keskula, J. Malloy, R. Keys, H. Zhang, and S. Jin, 2007, Iterative imaging for subsalt interpretation and model building: The Leading Edge, **26**, 1424–1428.

Shi, J. and J. Malik, 2000, Normalized cuts and image segmentation: Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence, **22**, 838–905.

Valet, L., G. Mauris, P. Bolon, and N. Keskes, 2001, Seismic image segmentation by fuzzy fusion of attributes: IEEE Transactions on Instrumentation and Measurement, **50**, 1014–1018.

Zahn, C. T., 1971, Graph-theoretical methods for detecting and describing gestalt clusters: IEEE Transactions on Computers, **C-20**, 68–86.