Interpreter aided salt boundary segmentation using SVM regression shape deformation technique

Problem Introduction

For the seismic imaging project in offshore oil and gas exploration, a good velocity model is a must in order to achieve the best seismic image quality. Accurate salt body geometry extraction is a key component in the velocity model building workflow. However, salt-body boundary picking is a very human-intensive procedure even with the help of currently available automatic picking tools.

It is very unlikely that there can be a super algorithm that can fully automates the boundary extraction procedure without sacrificing the result quality. Therefore this report addresses the problem of automatically and accurately segmenting the salt body in a series of neighboring seismic slices given only the accurate salt segmentation result on *one single* reference slice. This is achieved with a landmark based shape deformation technique plus SVM style regression. In this way, we can achieve the optimum trade-off between the amount of manpower and the quality of segmentation result.

Due to the large noise in seismic data, simple image processing filter plus local boundary tracker algorithm gives very poor result or fails easily; very often some parts of the salt boundary are just missing due to poor illumination or miscellaneous source of noise (see Fig 1). Human interpreter's inputs at these trouble regions are essential (Halpert, 2011).

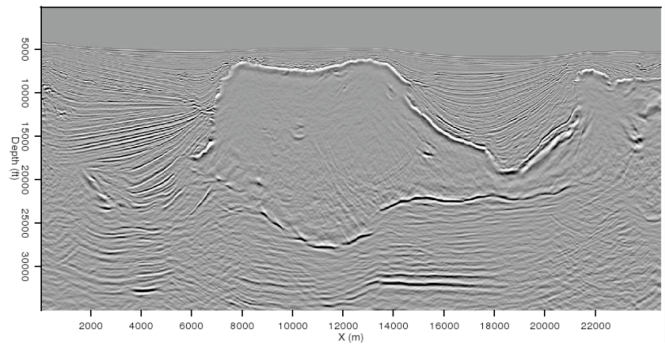
However as today's seismic imaging practices evolve to 3-dimensions (think that the 2-D images now becomes 3-D volume), it would be too time-consuming for the human to manually take care every single slice. Therefore the idea of manually segmenting only a small amount of "key" slices and then intelligently propagating these results to the entire volume comes into play. Here I use a land-mark based shape-deformation technique to propagate a single slice of manual segmentation result into its neighboring slices, thus yielding much better segmentation result than simply applying automatic methods. The design goal for such intelligent boundary propagation consists of two parts:

- First, I ask the new boundary to honor the *available* boundary information that can be extracted from the image.
- Secondly, the new boundary should preserve the shape information known from the manual segmentation input, such that the boundary will deform reasonably where we don't have well-defined boundaries.

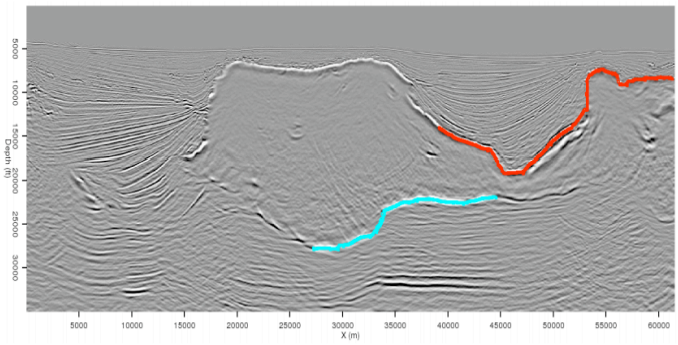
Methodology

Similar situations are encountered in the medical imaging field as well. My approach is based on the paper from Wang (2001) in ICCV'2001 (Please refer that paper for details). I will briefly describe this workflow in the following paragraphs:

Let me define the references image slice that has been properly segmented as the *template image*, and define the image that we want to propagate the segmentation result to as the *input image*. The segmentation result in the template image is characterized by a set of contours. For simplicity, we consider the case in which the template contains only one single closed contour. Extension to the case of multiple contours is straightforward.



(a)



(b)

Fig1. (a) A typical seismic image showing the salt body in the center. Some parts of the boundary are not well depicted due to the very limited image quality. (b) The same image shown with the human-interpreted boundary.

We represent the known contour as a set of landmark points, $V=\{V_1,V_2, \ldots, V_n\}$ where $V_i = (x_i,y_i)$. Then for each landmark V_i, the proposed method first identifies a set of possible corresponding landmark points $B_i = \{V_i^{(j)}, j=1,2,...,n_i\}$ on the input image, where $V_i^{(j)}=(x_i^{(j)},y_i^{(j)})$. Then conceptually the problem is solved in two major steps:

1. Identify the best landmark point v_i' from the landmark set B_i such that $V' = \{V_1',V_2', \ldots, V_n'\}$ located in or near the real object boundary in the input image.
2. Deform the prior shape V to match V' while keeping the general shape characteristics of V.

The cartoon below illustrates this landmark-searching step.

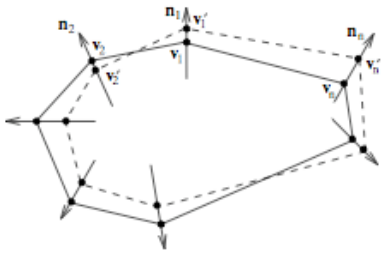


Fig2.Search for best landmarks on the input image based on the initial landmarks in the template (Wang, 2001).

In practices, it is difficult to determine the best landmark point v_i' in advance, and therefore we just choose randomly an element in B_i as the initial set V' and iterate this process during which we update the set V' such that V' more likely contains the correct corresponding landmarks.

The deformation step is formulated by finding the optimal solution to an objective function which takes into account both the goal of deforming the points in V into the current landmark set V' and the goal of preserving the prior shape V (use the bending energy formula from Bookstein (1989)). The objective function is shown below

$$\frac{1}{n}\sum_{i=1}^{n} Q(\mathbf{v}_i', \mathbf{t}(\mathbf{v}_i)) + \lambda\phi[\mathbf{t}], \qquad (*)$$

in which **t** defines the deformation as a mapping, i.e. **t**: (x,y)→(f(x,y),g(x,y))=(x',y'). Function Q describes the term that penalizes the mismatch between V' (the landmarks we found on the input image) and the mapping defined by **t**(V), such term corresponds to the first goal mentioned at the beginning of this paragraph. Function $\phi[\mathbf{t}]$ is a regularization term, which tries to force the mapping to **t** be smooth, in other words, preserving the global shape information of the original V.

We can see that these two terms are fighting with each other; therefore a **λ** parameter is added to balance these two parts. The optimization is solved using the classical SVM optimization technique, the fitting outliers are identified by the support-vectors, therefore we use this information to update the estimate of the best landmarks s et V'. I iterate the procedure above for a few iterations.

Experiment results

I test this algorithm on a 3-D seismic image cube from Gulf of Mexico survey. The cube is of discrete size 970X784X12 in depth(Z)/inline(X)/crossline(Y) directions respectively. I have the human-interpreted segmentation result in slice 1 only, which I use as the template image. I did initial processing on this segmentation result to extract the template landmarks as shown below.

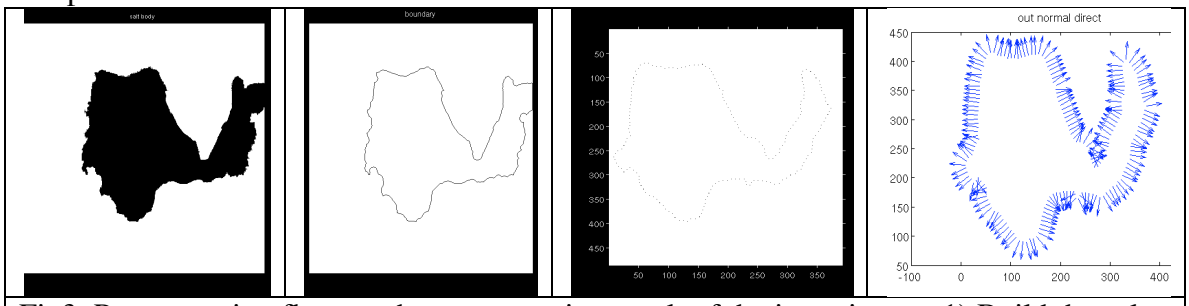


Fig3. Preprocessing flow on the segmentation result of the input image. 1) Build the salt-body mask. 2) Extract the boundary. 3) Subsample to a list of landmarks. 4) Find the out-normal directions for each landmark on the contour.

To find candidates for the set B_i, I overlay the landmark on the input image (neighboring slice), then I search along the normal direction for certain image features which might suggest the position to be a part of boundary. Here I just use a very simple criterion, I choose the image locations that the image envelope amplitude local maxima lie as the boundary point candidates.

Then I run the SVM optimization for a few iterations, within each iteration I identifies all the support vectors (which corresponds to the fitting outliers); for each support vector index i, I try to use other candidates in B_i set such that the fitting $|h_i - \gamma_i|$ gets improved.

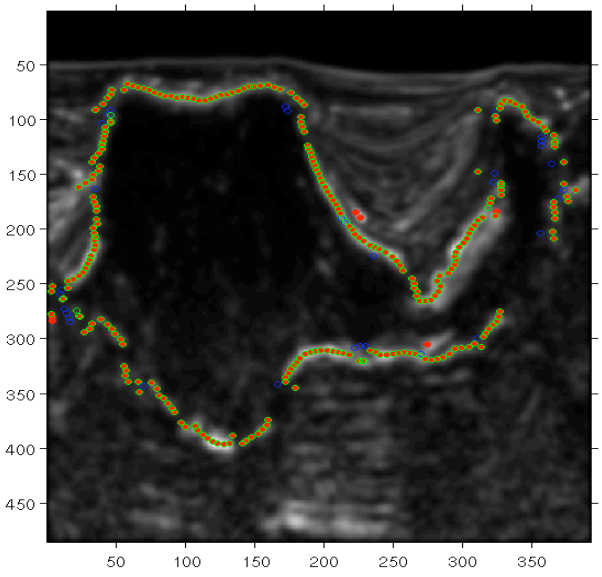


Fig5. The update of set V' in one iteration. Blue indicates all support vector points, red shows the original points in V', green shown the updated V' set by replacing the badly fitting point v_i' with better fitting candidates in B_i.

Finally I deform all 12 slices one by one, from slices close to the template slice to slice far from the template. In figure 6, I show the comparison between the obtained deformed boundary and the boundary found by automatic methods. The improvement is very prominent.

Conclusion

In this paper, I propose to improve the salt-body segmentation for 3-D seismic images by deforming the accurate boundary on human-interpreted slice into the neighboring slices. The deformation not only honors the available boundary information on the input slice, but also preserves the shape information from the template slice. The numerical examples show very promising results.

References

- Song Wang; Weiyu Zhu; Zhi-Pei Liang; , "Shape deformation: SVM regression and application to medical image segmentation," Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, vol.2, no., pp.209-216 vol.2, 2001
- Halpert, A., R. G. Clapp and B. L. Biondi, 2011, Interpreter guidance for automated seismic image segmentation: Presented at the 74th Annual International Conference and Exhibition, EAGE.
- Bookstein, F.L. , Principal warps: Thin-plate splines and the decomposition of deformations. IEEE Trans. PAMI, 11:567– 585, June 1989.

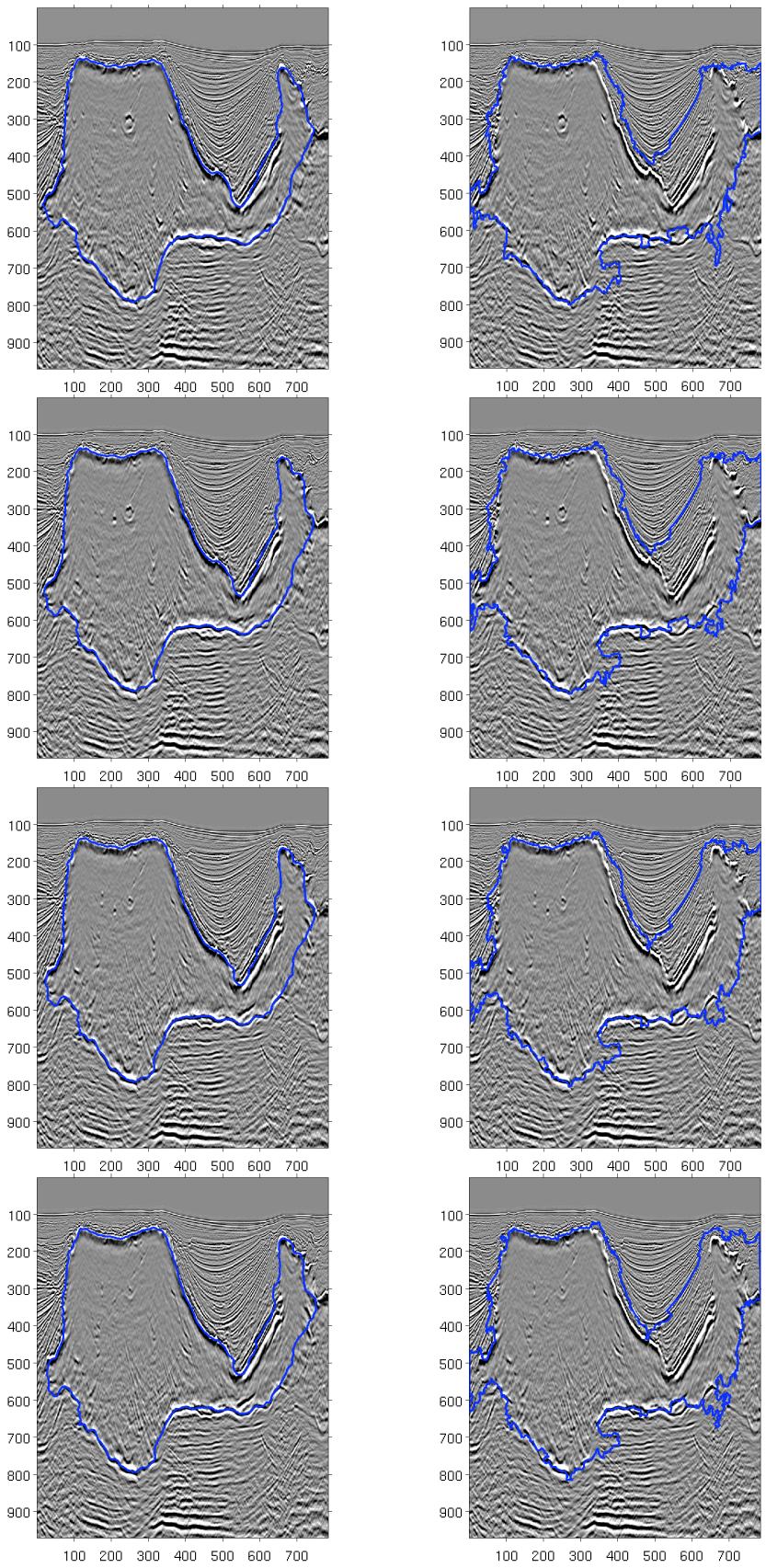


Fig 6. The segmentation result for Slice NO. 1,4,8,12 using the boundary deformation (left column) and using the simple automatic methods (right column).

Appendix

Math details for solving the optimization (you can safely skip it if you are not interested)

With regard to the actual form of the objective function in (*), particularly I choose function **Q** to be the ε-insensitive L1 norm, as the following

$$= \begin{array}{l} ||\mathbf{v}_i' - \mathbf{t}(\mathbf{v}_i)||_\epsilon \\ \begin{cases} 0 & \text{if } ||\mathbf{v}_i' - \mathbf{t}(\mathbf{v}_i)|| \le \epsilon \\ ||\mathbf{v}_i' - \mathbf{t}(\mathbf{v}_i)|| - \epsilon & \text{else.} \end{cases} \end{array}$$

.

Since new landmarks in V' is searched along the shape's normal direction $\mathbf{n}_i$. We confine that the desired mapping **t**(V) should map along $\mathbf{n}_i$ as well, i.e.

$$\mathbf{t}(v_i) = v_i + \gamma_i \mathbf{n}_i. \quad (1)$$

Since points in V' is found along normal directions of V as well, we have $v_i' = v_i + h_i \mathbf{n}_i$, then the previous objective function can be written as

$$\min_{\mathbf{t},\boldsymbol{\gamma}} \Phi(\mathbf{t},\boldsymbol{\gamma}) = \frac{1}{n}\sum_{i=1}^{n} |h_i - \gamma_i|_\epsilon + \lambda\phi[\mathbf{t}]$$

subject to constrain (1).

As for $\phi[\mathbf{t}]$, we choose the so called *bending energy* term, it is defined as

$$\phi[\mathbf{t}] = \iint_{-\infty}^{\infty} (L(f) + L(g))dxdy$$

,

where $L(\cdot) = (\frac{\partial^2}{\partial x^2})^2 + 2(\frac{\partial^2}{\partial x \partial y})^2 + (\frac{\partial^2}{\partial y^2})^2$. The nice thing about choosing such bending energy term, is that we know in advance, given all the mapping **t** that satisfies constrain (1), the thin-plate spline interpolation will minimize the bending energy (Bookstein, 1989), which takes the form: **t** = (f,g), and

$$\begin{cases} f(\mathbf{v}) = a_1 + a_2 x + a_3 y + \sum_{i=1}^{n} c_i K(\mathbf{v}, \mathbf{v}_i) \\ g(\mathbf{v}) = b_1 + b_2 x + b_3 y + \sum_{i=1}^{n} d_i K(\mathbf{v}, \mathbf{v}_i). \end{cases}$$

K is the well-known thin-plate kernel function.

As shown in Wang(2001), we can also represent $\phi[\mathbf{t}]$ with the vector $\gamma = \{\gamma_i : i = 1,.2,..n\}$, therefore this variational problem (optimization with variables as functions) turns into one much simpler numerical convex optimization problem. Then we just try to find the optimal $\gamma = \{\gamma_i : i = 1,.2,..n\}$ that

$$\min_{\boldsymbol{\gamma}} \Phi(\boldsymbol{\gamma}) = \frac{1}{n}\sum_{i=1}^{n} |h_i - \gamma_i|_\epsilon + \frac{\lambda}{8\pi}(\hat{\mathbf{x}}^T \mathbf{L} \hat{\mathbf{x}} + \hat{\mathbf{y}}^T \mathbf{L} \hat{\mathbf{y}})$$

,

here $\hat{\mathbf{x}}$,$\hat{\mathbf{y}}$ is the vector representation of the first and second coordinates of the points in set **t**(V) (keep in mind that $\mathbf{t}(v_i) = v_i + \gamma_i \mathbf{n}_i$).

Using the standard SVM technique, we can instead solve the dual problem according to the K.K.T. condition. It ended up as being a *standard quadratic programming* problem with both upper and lower bounds (see Wang(2001)).