Reimaging Shallow Structure

Greg DePaul & Jeremy Wood

***Abstract*—We perform smoothing over the raw velocity spectrum data, and then apply methods of unsupervised classification to emphasize features of nonuniformity. We then create a heuristic-driven graphical overlay for accelerating subsurface fault-line/mineral deposit identification based on reflection seismology.**

I. Introduction

Seismic Reflection is the method of sending a pulse (an acoustic pressure wave) into the ground and detecting the reflected energy using an array of sensors. Using the time it takes for the repeated pulses to return to the surface, seismic reflection reconstructs the stratigraphic features within the crust. These can be reconstructed thanks to the difference in acoustic impedances between layers at the interface between different types of rock or other subsurface material. This type of processing yields, among other things, measurements of trapped hydrocarbons and natural gas within the shallow crust as they disrupt reflected waves.

However there are issues with reconstructing the structure directly. Due to noisy reflection, seismic pulse exploration only roughly correlates with the underlying subsurface structure. There are three types of noise: Natural Noise, Cultural Noise, as well as Secondary Reflections. We want to develop upon this scheme to reduce the noise over repeated pulses and visually reconstruct the shallow geological structure. Furthermore, we hope to use this noise-attenuated dataset to make progress towards characterizing chemical deposits. Our goal, after inputting our data set, is to develop an algorithm with the ability to distinguish interesting features of the shallow structure and appropriate coloring, based on seismic data. Currently, geologists spend copious time examining small patterns in images such as those presented in Fig. 1 and Fig. 2. By visually highlighting non-standard formations while appropriately reducing the visual significance of repetitive geological features, we hoped to contribute towards a less time-intensive exploration process. Automating the characterization of irregular underground features would speed up the processing and tedious analysis of geophysical exploration.

II. Description of Data Set

Our data comes from the National Centers for Environmental Information (under the National Oceanic and Atmospheric Administration). Most of the data that we used was collected in the late twentieth century by government researchers. The data itself was all contained within SEGY formatted binary files. SEGY is a format developed for the storage and transmission of geophysical data. Within this context, the SEGY files contain the ”traces” acquired from sampling the geophones' input (i.e. the reflected energy pulses mixed with background noise) over a period of time after the pulse has been sent out. Each trace consists of the received input for a single geophone after a single pulse. These trace collections represent completely raw measurements covering two axes: recorded signal strength and time. In order to be used to accurately depict subsurface structure, this data normally must go through filtering, signal processing, adjustment, and merging. All of these processes are intended to reduce noise, eliminate background interference/strengthen the pulse strength, and migrate the data from a representation over time to a representation over space (and thus enable mapping). Some of the most basic processing leads from images like that in Figure 1 to the processed image in Figure 2. Ours had to be congregated and manipulated (mostly in Matlab) to get to Figure 1 and then further processed before it could be interpreted. Figure 2 represents a noisy depiction of subsurface features and layers. What we hoped to help distinguish were faults and other anomalies that could indicate chemical deposits. Fault lines usually present themselves as sheered formations within the seismograph. Thus we hoped to highlight portions of these images wherein horizontal striations become disrupted–either forming an incline or becoming random noise. One key insight was realizing that though not all disruptions of horizontal striations are significant, there are few to no significant portions of seismic data that are normal horizontal striations. (Figure 3a).

It is this preprocessed data that the rest of our unsupervised learning will take place on. Notice that visually Figure 2 is seemingly uniform. However, upon applying machine learning methods, we would hope this becomes visually revealing of its structure.

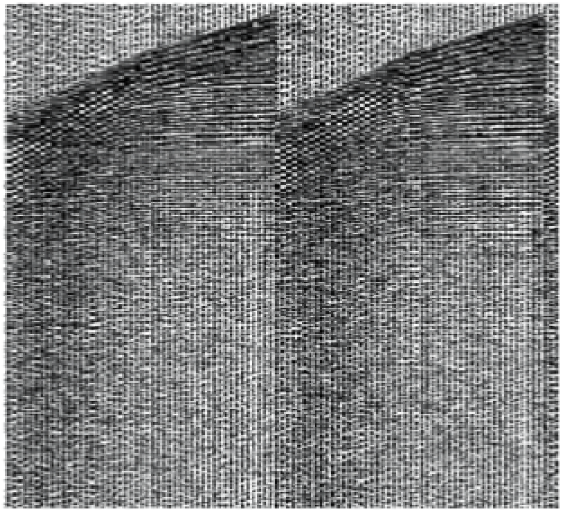


Fig. 1. Unprocessed Segy Data

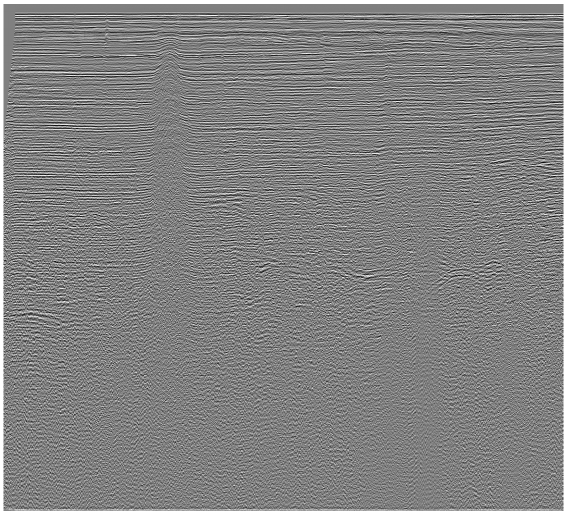


Fig. 2. Imaged Shallow Crust after processing Segy Data

III. Previous Work

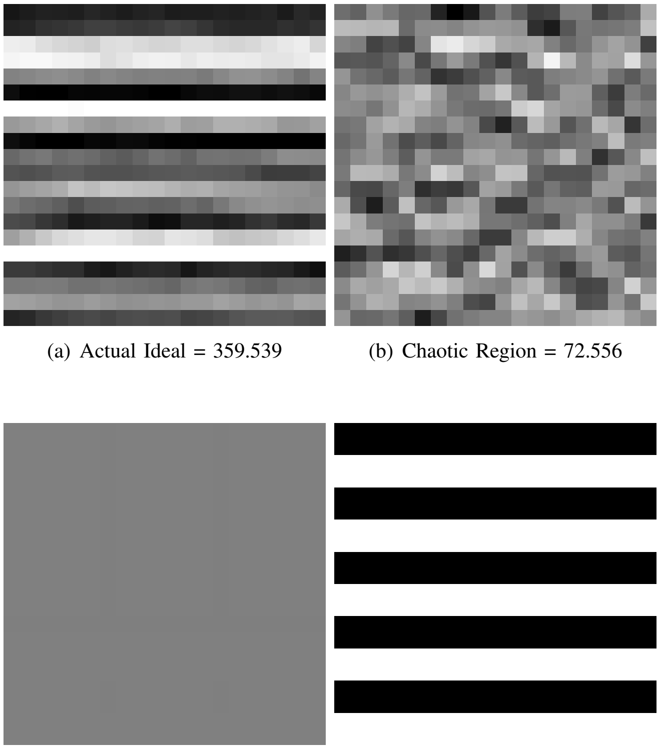
Much of previous work in applying Machine Learning to Seismic Reflection take a rather different approach to solve this problem. Papers, like the one written by Evans and Wecht, utilize a supervised learning algorithm over synthesized data in order to develop a function that takes in rarefaction waves and produces a velocity model that highlights faults. Our approach on the other hand relies on detecting anomalies in real data sets using unsupervised learning. Though previous worked served as useful reference material, much of the work we did was independent of previous attempts at analyzing seismic reflection data.

IV. Approach to Solution

Data Input: Given that SEG-Y is a non-trivial yet standard format, we eventually used prefabricated packages in Python to retrieve the trace data from disparate files. This raw data can be observed in Figure 1 above.

Preprocessing: Convert the seismic reflection data into a digestible array of velocity data, as in Figure 2. In our case, we chose to represent the data as an image file to ease input/output. This array is then discretized into 20 by 20 boxes that will provide us an outlet for our metric. We used the Python Imaging Library (PIL/Pillow) to save, load, and show images.

Metric Definition: The metric we developed was based on a spectrum of variances. These variances are measured on a 20x20 pixel box (m=20). We developed this idea based on the overwhelming repetition of these patterns within observed crust models. Specifically, flat, contiguous crust tends to exhibit the pattern in the first image of Figure 3. On the other hand, as layers interact with each other, we start to see a pattern that more closely resembles the second image in the same figure. We wish to perform clustering to investigate the overall connectedness of this space. Graphically, our metric distinguishes regions as depicted in Figure 3. Notice that the homogenous (notably: vertically homogeneous) pattern yield a metric value close to zero. On the other hand, as you increase the variance over vertical means of the horizontal colors, you begin to approach higher numbers. As a sanity check, you can see we included a fourth pattern not actually found within our data that serves as a theoretical maximum of our metric.



(a) Actual Ideal = 359.539

(b) Chaotic Region = 72.556

(c) Homogenous Region = 0.923

(d) Theoretical Ideal = 570.197

Fig. 3. Regions selected from Figure 2 that represent areas of interest and their respective metrics.

Mathematically, we define the metric as follows. Initially we consider the horizontal mean, over an $m \times m$ pixel square:

$$\mu_{xy} = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left\| v_{(x+i,y+j)} \right\|$$

Where $v_{(x+i,y+j)}$ is the RBG vector of the pixel $(x+i, y+j)$. This makes sense to use since were are working in black and white data. Then we can define our metric as a sort of variance:

$$\sigma_{xy}{}^2 = \sum_{i=1}^{m} \left(\mu_{xy} - \sum_{j=1}^{m} \left\| v_{(x+i,y+j)} \right\| \right)^2$$

This metric now allows us to compare regions together more effectively. We can now use K-Means to cluster regions of similar weights, and converge to something visually revealing.

Clustering Algorithm:

1. Perform a motion estimation algorithm over the data set by progressing a box metric over the entire set.
2. Each box then assigns a metric over that particular region
3. Regions of similar metrics are then clustered together.
4. Repeat until the metric labeling becomes stable. We used the scikit-image library to run K-Means over the boxes.

Postprocessing: The identified features labels for each box are then assigned colors randomly and these colors are drawn over their respective boxes. These boxes are then unpackaged to recreate a pixel-discretized version of the original image. This colorized discretization is then overlaid to better emphasize features of interest in the original image. These colors should make apparent the features obfuscated by the original trace image.

V. Results

Upon running our algorithm, the model in Figure 2 results in our overlay depiction in Figure 4. Examining, you'll notice that what was once a homogenous looking structure is now categorized into three separate layers. Immediately one can see that a chaotic portion of the middle layer spikes into the top layer. This gray area highlights a structural abnormality, which can be noted and further examined.

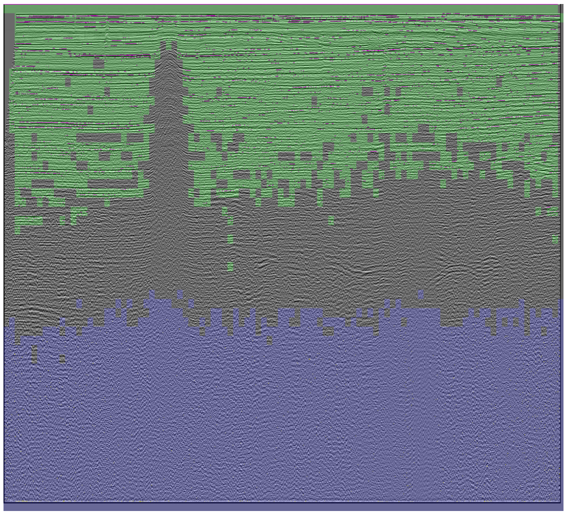


Fig. 4. Feature overlay applied to model in Figure 2

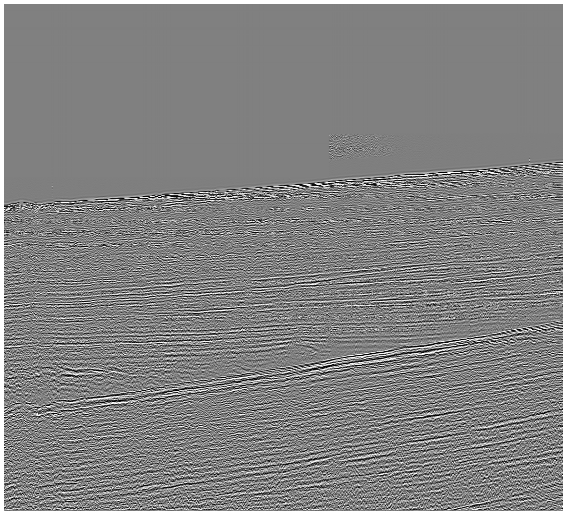


Fig. 5. Slight fault example

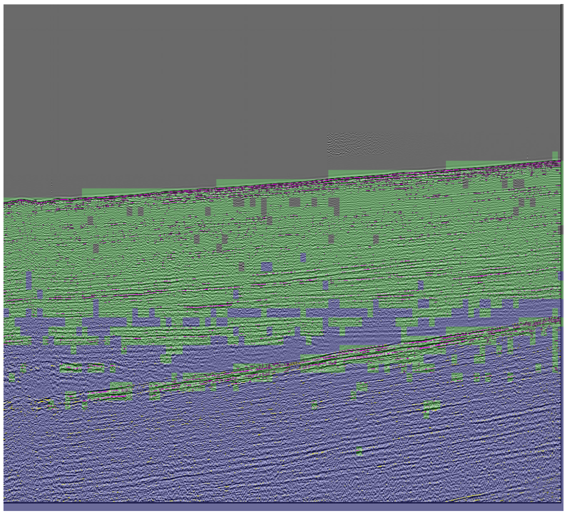


Fig. 6. Overlay applied to slight fault example

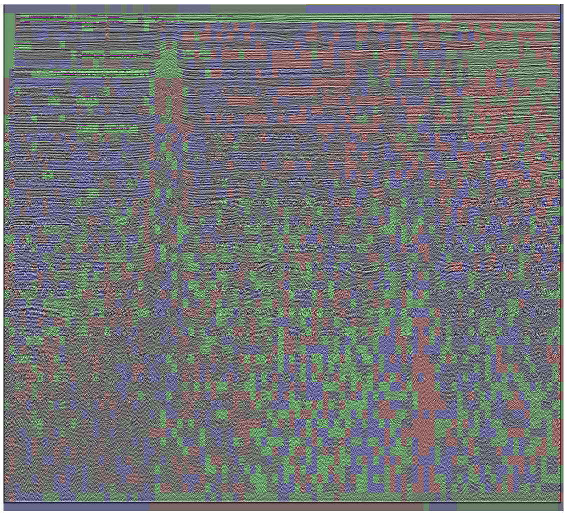


Fig. 7. Checkerboard effect due to pattern over-recognition

Another example of this algorithm is the case of a very slight (less than $45°$) fault, as in the crust in Figure 5. Our algorithm creates the overlay shown in Figure 6. This overlay highlights that there are two layers which have possibly collided and created structural anomalies at the boundary between the two.

VI. Discussion

Applying the clustering algorithm requires assuming a certain number of clusters that should be made apparent. In all our examples we typically select 3 or 4 clusters of patterns. This number should be viewed as describing how many layers and homogeneous portions you expect to find with the clustering algorithm. Selecting more patterns results in a ”checkerboard” behavior, simply because you have allowed the algorithm to distinguish so many patterns that nearly every component is attributed as an interesting feature. This is evident in Figure 7 which represents an overlay over Figure 2 with the assumption of 8 clusters. This example shows that the number of clusters must be picked carefully. However, for datasets over successive and physically proximal sections of earth this layer estimation should hold relatively constant. Thus it can be set early on and then reused without tuning.

VII. Conclusion And Continuation

We were able to apply clustering techniques to identify features of interest in shallow crust models based on a custom, heuristic based metric. This type of clustering relies heavily on the geometry of the velocity models, specifically the uniform striations between layers that indicate contiguous layering. These types of patterns can then be clustered together in an effective way to indicate relative locations of interesting structures in the crust that may otherwise go undetected or require tedious work to uncover.

Currently our algorithm relies on the motion estimation to lie exactly on the block grid. In the future, it would be better to allow more maneuverability. It would also be beneficial to include a technique that could handle sloped inclines, as evident in Figure 5.

VIII. References

- Lillie, R. J. (1999). Whole earth geophysics: An introductory textbook for geologists and geophysicists. Upper Saddle River, N.J: Prentice Hall. Pages 118 through 127.
- Evans, Wecht. Machine Learning for Seismic Reflection Data. Retrieved from Harvard Faculty of Arts and Sciences: http://people.fas.harvard.edu/~wecht/Kevin_J_Wecht/Data_Analytics_files/Final_Report.pdf
- NOAA. (1977). ECT14-17 [Digital SEG-Y]. Retrieved from https://www.ngdc.noaa.gov/mgg/trk/trackline/coral_seal/ect14-17/seismics/data/digital/