FAULT DETECTION IN SEISMIC DATASETS USING HOUGH TRANSFORM

Zhen Wang and Ghassan AlRegib

Center for Energy and Geo Processing - CeGP
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA, 30332-0250
{zwang313, alregib}@gatech.edu

ABSTRACT

In this paper, we present a semi-automatic algorithm to detect faults in seismic datasets using Hough transform. As a multistage approach, our method first highlights the likely fault points from the discontinuity map of one seismic section. Hough transform is then applied to detect faults features. Considering geological constraints of faults, false features are removed using a double-threshold method. Then, we get an initial fault line by connecting the remaining faults features. In the last stage, by incorporating the discontinuity information from step one, we tweak the initial fault line to obtain more accurate and reliable results. Our experimental results show that our method can delineate the fault lines in seismic sections more accurately than a state-of-the-art method.

Index Terms— Hough transforms, fault detection, feature extraction, seismic imaging

1. INTRODUCTION

A Fault, a common geological structure, is formed by a displacement between neighboring tectonic plates. In some cases, faults seal the porous reservoir rocks and lead to the formation of petroleum reservoirs. Hence, the detection of faults in seismic volumes is of great importance for the identification of reservoir regions. The accuracy of fault detection is strongly tied to the cost of the planned drilling and the boreholes. For several decades, fault detection was implemented manually and its accuracy depended on the experience of interpreters. For gigantic amount of seismic data, manual interpretation is very time-consuming. Therefore, the use of computer programs to detect faults in large seismic volumes has become an active research area for several years. Although it is difficult to implement fully automatic fault detection in practice, many companies, such as Schlumberger [1] and Exxonmobil, have built their interactive software platforms to label faults semi-automatically.

Faults are caused by rock movements, which cause discontinuities in horizons. Hence, detecting such discontinuities is the main challenge in identifying faults in seismic datasets. Several algorithms were proposed in the literature to detect discontinuities in horizons using certain seismic attributes. Examples of such attributes include coherence [2], variance [3], curvature [2] or gradient amplitude [4, 5]. Depending on the seismic attributes alone does not fully detect faults with a reasonable accuracy that is expected by the interpreters. Therefore, another class of techniques have been proposed to extract fault information from the calculated seismic attributes. For example, Gibsen et al. [6] proposed a multistep workflow of fault surface detection, which first grouped the highlighted fault points in local planar patches and then merged these patches into large fault surfaces. Cohen et al. [7] applied multiple directional filters to enhance the contrast of the discontinuity cube and then used skeletonization to extract the thinning fault surfaces. More recently, a class of fault detection techniques were proposed and are based on the concept that faults can be described as structured edges. For example, Hough transform was proposed to delineate the fault lines in time slices [8] and was extended to detect fault surfaces in its cascade form [9]. Although the results in these papers are not strong, they serve as a very good start to employ image processing algorithms to detect seismic attributes. More recently, Hale in [10] proposed an automatic fault detection method based on directional Gaussian filter. In this method, the semblance is used to generate a discontinuity map. Then, a directional Gaussian filter is used to refine the discontinuity map. Finally, the maximum discontinuity value within a window is chosen to represent a fault point. The whole operation in [10] is fully automated and efficient.

In this paper, we propose a reliable fault detection algorithm that is based on Hough transform. The block diagram of our proposed method is shown in Fig. 1. First, the likely fault points are highlighted in a binary image by thresholding the discontinuity map. Second, we apply Hough transform to the binary image to detect fault features. In the third step, we propose a method to eliminate false features based on geological constraints. In the final step, we tweak the fault lines using the discontinuity information obtained from the first step.

In Section 2, we explain the proposed methodology in details. Section 3 illustrates the experimental results while we conclude the paper in Section 5.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

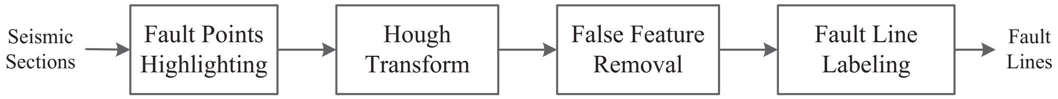


Fig. 1: Block diagram of fault detection method based on Hough transform

2. THE PROPOSED METHOD

2.1. Fault Points Highlighting

Discontinuities in horizons are the most important characteristic of faults that can be used to highlight the likely fault points. To measure discontinuities in seismic sections, we calculate the semblance attribute that was proposed by Marfurt et al. [11] and it outperforms earlier algorithms by involving the local dip information. For every point in a seismic section, its discontinuity value can be obtained from an analysis window that is oriented at the same dip with the local horizon. In such a square window with size $2r+1$, discontinuity value $D(x,z)$ can be computed as:

$$D(x,z) = \left| \ln \left[\frac{\sum\limits_{j=-r}^{r} \left(\sum\limits_{i=-r}^{r} S(x+i,z+j) \right)^2}{(2r+1) \sum\limits_{i=-r}^{r} \sum\limits_{j=-r}^{r} S(x+i,z+j)^2} \right] \right|, \quad (1)$$

where $S(x,z)$ is the intensity of the seismic signal at point (x,z). In seismic sections, x and z represent the coordinates on crossline and depth direction, respectively. The $\ln(\cdot)$ operator used here increases the contrast between high and low discontinuities, which makes the selection of the likely fault points more reliable. Greater value of $D(x,z)$ means a higher probability that the corresponding point belongs to a fault. If $D(x,z)$ is close to zero, the point is assumed to be located on a horizon. Therefore, we apply a threshold T_0 to the discontinuity map in order to determine the likely fault points. This thresholding operation is given as follows:

$$B(x,z) = \begin{cases} 1 & \text{if } D(x,z) \geq T_0 \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

The choice of T_0 is a designed parameter that can be determined empirically. After the thresholding step, a binary image is obtained where the likely fault points are highlighted as white pixels.

2.2. Hough Transform

Hough transform [12] as a feature extraction technique has been widely used in the field of digital image processing to detect edges. In general, points in an image space can be transformed to parameters of curves in a parameter space and vice versa with the following transform equation:

$$x \cos\theta + y \sin\theta = r. \quad (3)$$

As illustrated in Fig. 2, r and θ are constant for any point on the blue line in the image space. On the other hand, x and y are constant for any point on a curve in the parameter space.

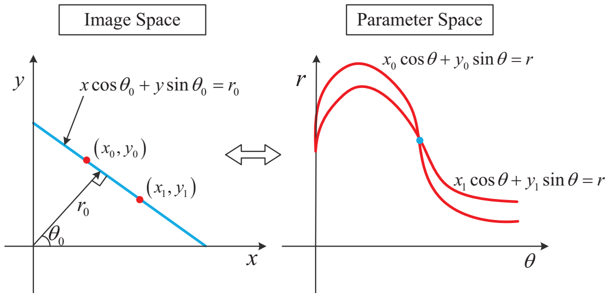


Image Space
Parameter Space
y
x
r
θ
$x\cos\theta_0 + y\sin\theta_0 = r_0$
(x_0, y_0)
(x_1, y_1)
r_0
θ_0
$x_0\cos\theta + y_0\sin\theta = r$
$x_1\cos\theta + y_1\sin\theta = r$

Fig. 2: Hough transform between image and parameter space

Henceforth, a line in the image space can be represented by the intersection of curves in the parameter space. In this paper, we use this idea to detect fault features in the form of lines from the highlighted points by identifying points in the parameter space where the largest number of lines intersect.

Generally, detection of fault lines based on Hough transform is an interactive process where the user could specify the characteristics of the output. For example, the user could specify the number of lines, the minimum length of these lines, and the slope interval. Although these parameters affect the output quality, users do not have to spend too much time on tweaking them because of the removal of the false features in the following section.

2.3. False Feature Removal

It is inevitable to have false features in the results from the previous step. To overcome this issue, we propose a double-threshold method to remove all false features. Fig. 3(a) illustrates two examples of the output of a Hough transform step. The first example is inside the dashed box while the second example is in the solid box. Based on geological constraints, the lines inside these two boxes are considered false. The one in the dashed box is an *outlier* because it is isolated from other detected lines or features. The lines in the second example in the solid box represent *neighboring group* where the lines are very close to each other, which is geologically impossible. One method to recover from the artifact of *neighboring group* is to combine all lines within a *neighboring group* into one line.

The identification of the false features illustrated in Fig. 3(a) depends on a function of the spatial relationship between lines that are produced by the Hough transform. In this paper, we define two such functions, *absolute distance (AD)* and *lateral distance (LD)*. *Absolute distance* represents the spatial distance between two lines, which can be computed as:

$$AD = \|\mathbf{l}_i - \mathbf{l}_{i-1}\|_2^2, \quad (4)$$

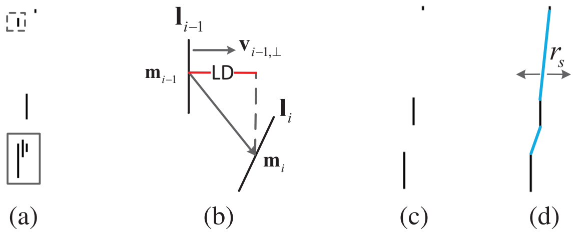


Fig. 3: (a) Features of faults detected by Hough transform. (b) Illustration of *lateral distance* (LD). (c) Features after removing false ones. (d) Initially labeled fault line by connecting features in (c).

where $\mathbf{l}_i, i = 1, 2, \cdots, n$ are the 2×2 matrices containing the starting position $(x_{b,i}, z_{b,i})$ and the ending position $(x_{e,i}, z_{e,i})$ of lines and n is the total number of distinct lines. Meanwhile, the length of these lines $L(\mathbf{I_i})$, $i = 1, 2, \cdots, n$, can be calculated as: $L(\mathbf{I_i}) = \sqrt{(x_{b,i} - x_{e,i})^2 + (z_{b,i} - z_{e,i})^2}$. The ordering of these lines depends on their vertical spatial positions.

In the definition of the *lateral distance*, as shown in Fig. 3(b), vector $\mathbf{m}_i = (\frac{x_{b,i}+x_{e,i}}{2}, \frac{z_{b,i}+z_{e,i}}{2})$ represents the midpoint of line $\mathbf{I}_i$. The lateral distance can be derived from the projection of the vector connecting $\mathbf{m}_i$ and $\mathbf{m}_{i-1}$ on the direction perpendicular to $\mathbf{I}_{i-1}$ as shown in the following equation:

$$LD = | (\mathbf{m}_i - \mathbf{m}_{i-1}) \cdot \mathbf{v}_{i-1,\perp} |, \tag{5}$$

where $\mathbf{v}_{i-1,\perp}$ is the unit vector perpendicular to $\mathbf{I}_{i-1}$. Using the two quantities AD and LD, we apply a double-threshold method to filter out the false features. The double-threshold method is given in the following pseudo code.

Algorithm 1 False feature removal

```
for $i \leftarrow 1$ to $n$ do
    if $LD \geq T_1$ then
        $\mathbf{l}_i \leftarrow \mathbf{l}_{i-1}$
    else
        if $AD \leq T_2$ then
            $\mathbf{l}_i \leftarrow \underset{\mathbf{x} \in \{\mathbf{l_i}, \mathbf{l_{i-1}}\}}{\arg\max} \ L(\mathbf{x})$
        end if
    end if
end for
```

T_1 is the threshold for LD and T_2 is the threshold for AD. If LD is greater than T_1, then $\mathbf{l}_i$ is far away from $\mathbf{l}_{i-1}$ and should be discarded. Similarly, if AD is less than T_2, then $\mathbf{l}_i$ and $\mathbf{l}_{i-1}$ are located very close to each other and they should be merged into a longer line.

2.4. Fault Surface Labeling

After removing the false features, the remaining features construct the main part of faults, as shown in Fig. 3(c). Here, we define the matrix $\mathbf{P}_0 = (\mathbf{x}_0, \mathbf{z}_0)$ to describe the positions of all points in these features, where $\mathbf{x}_0$ and $\mathbf{z}_0$ respectively contain the coordinates on crossline and depth direction. Therefore, in Fig. 3(d), by connecting the detected features successively, fault line is initially labeled and denoted as $\mathbf{P}_c = (\mathbf{x}_c, \mathbf{z}_c)$. The labeled result is not accurate to describe the details of the fault. To improve the accuracy, we utilize the discontinuity map and re-label the fault.

The position of a point i in $\mathbf{P}_c$ can be denoted as $\mathbf{P}_c(i) = (\mathbf{x}_c(i), \mathbf{z}_c(i))$, $i = 1, 2, \cdots$. Searching along the crossline direction with radius r_s illustrated in Fig. 3(d), a group of points $\mathbf{P}_m = (\mathbf{x}_m, \mathbf{z}_c)$ corresponding to the local maximum discontinuity values are identified. We denote these points as $\mathbf{x}_m(i)$ and we calculate these points using the following expression:

$$\mathbf{x}_m(i) = \underset{x \in [-r_s + \mathbf{x}_c(i), \mathbf{x}_c(i) + r_s]}{\arg\max} D(x, \mathbf{z}_c(i)). \tag{6}$$

Now, for each crossline, we have two points that are candidates to be fault points. One point, $x_c(i)$, which lies on the line in Fig. 3(d). The second point , $x_m(i)$, is obtained from the discontinuity map using Eq. (6). The next step is to decide the location of the fault point within the interval $[x_m(i), x_c(i)]$. Therefore, we need to measure the relative influence of $\mathbf{x}_c$ and $\mathbf{x}_m$ using the following objective function:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\arg\min} \ \lambda_1 \|\mathbf{x} - \mathbf{x}_c\|_2^2 + \lambda_2 \|\mathbf{x} - \mathbf{x}_m\|_2^2, \tag{7}$$

where λ_1 and λ_2 are the weights of $\mathbf{x}_c$ and $\mathbf{x}_m$ in the objective function and $\hat{\mathbf{x}}$ is the optimal labeling outcome. The resulting $\hat{\mathbf{x}}$ will have a zig-zag shape, which does not match a true geological structure. Therefore, we apply a smoothing filter to $\hat{\mathbf{x}}$ in order to filter out the high frequency components.

The next step is to combine the original strong fault features shown in Fig. 3(c) and the refined features of the blue lines which are shown in Fig. 3(d). We merge the smoothed result $\hat{\mathbf{x}}_s$ and the detected features $\mathbf{x}_0$, to obtain the optimized fault line per the following equation: $\mathbf{P}_d = (\mathbf{x}_d, \mathbf{z}_c)$:

$$\mathbf{x}_d(i) = \begin{cases} \mathbf{x}_0(k) & \text{If } \exists k : \mathbf{z}_c(i) = \mathbf{z}_0(k) \\ \hat{\mathbf{x}}_s(i) & \text{otherwise} \end{cases}. \tag{8}$$

3. EXPERIMENTAL RESULTS

The 3D seismic dataset used in our experiment is from the Netherlands offshore F3 block acquired in the North Sea [13]. We focus on the seismic volume containing long and apparent faults located in the inline range from 200 to 300, the crossline range from 700 to 1200 and the time range from 400ms to 1100ms.

The example in Fig. 4 demonstrates the seismic section Inline 256 and its corresponding discontinuity map, where the colors of red and light blue represent higher discontinuity in contrast to the dark blue ones for the continuous horizons. The likely fault points can be highlighted after thresholding the discontinuity map with $T_0 = 0.9$ and the result is shown in the binary image in Fig. 4(c). Next we apply Hough transform to Fig. 4(c) in order to detect the fault features from the highlighted fault points with the specified number of lines 20, line minimum length 2 and line slope interval $[80°, 100°]$. Fig. 5(a) illustrates the detected fault features in the form of the white line segments, which are distributed around the likely fault region. Here, the region where faults appear is magnified for a better illustration. Because of the appearance of false features, it's essential to remove them by using the double-threshold method. In this case, we selected $AD = 11$ and $LD = 4$. After removing the false features, the remaining ones in Fig. 5(b) are part of the actual fault. According to Eq. (7) with $\lambda_1 = 0.2$ and $\lambda_2 = 0.8$, the optimized fault line $(\hat{\mathbf{x}}, \mathbf{z}_c)$ has a zig-zag shape shown in Fig. 5(c). We further smooth the detected fault line to obtain a line $(\hat{\mathbf{x}}_s, \mathbf{z}_c)$ illustrated in Fig. 5(d). Finally, by combining $\hat{\mathbf{x}}_s$ and the fault features $\mathbf{x}_0$ in Fig. 5(b), we delineate the fault lines in the seismic section, as shown in Fig. 5(e).

In Fig. 5(f), we compare the white fault lines detected using our proposed method with the manually labeled ground truth in red. The two lines almost overlap at most locations except for the mismatch located mainly at the top of the image. Furthermore, Fig. 5(g) shows the white fault line detected by the approach in [10]. Comparing our method with the one proposed in [10], we see that our method produces more accurate results. To quantitatively measure the difference between the detected faults and the ground truth, we propose a distance function as $dist = \sum_{i=1}^{N} |\mathbf{x}_d(i) - \mathbf{x}_t(i)|^2$, where $\mathbf{x}_d$ and $\mathbf{x}_t$ respectively correspond to the x coordinates of N pixels in the detected fault lines and the ground truth. For Inline 256, the corresponding distance of the proposed method is 316 which is 15% less than the distance of 379 for the detected result in [10]. The main reason is the step where we remove false features, which is one of the major contributions of this paper.

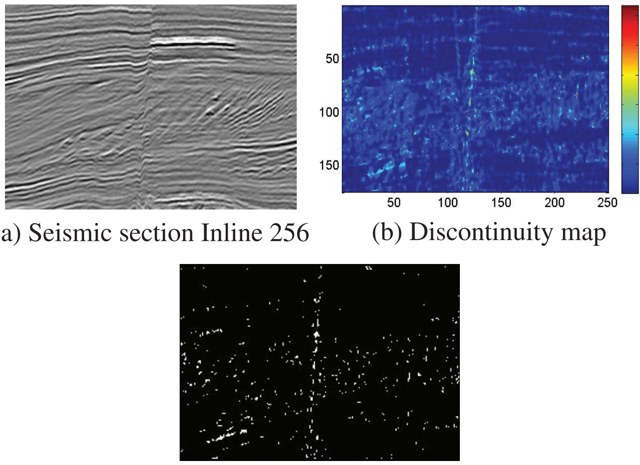
We further test the robustness of our method by applying the same method with the same parameters to Inline 246 and Inline 268. The results are shown in Fig. 6. Clearly, we can still delineate the faults with a very high accuracy.

4. ACKNOWLEDGEMENT

We would like to thank Prof. Dave Hale for sharing his method's codes. We also would like to thank Dr. Muhsin Wei for his observations in the early stages of this work.

5. CONCLUSION

In this paper, we proposed a semi-automatic algorithm for fault detection using Hough transform, which determines fault features from a set of candidate fault points. After the removal of false features, the remaining fault features act as the main parts of the entire fault. Finally, we tweak fault lines to obtain a more accurate result by the combination of the detected features and the discontinuity map. Experimental results show that our method has a better accuracy performance than the state-of-the-art method. Our method requires three threshold values that are designed parameters and dataset dependent. In a future work, we plan to automatically calculate these values to make our method fully automatic.

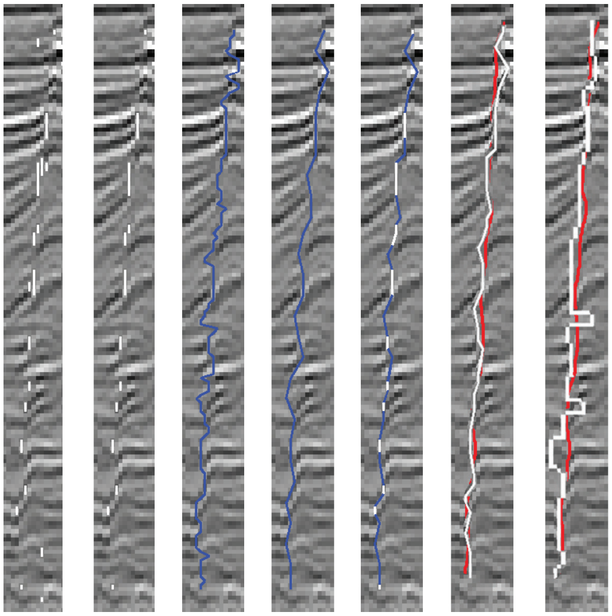


(a) Seismic section Inline 256

(b) Discontinuity map

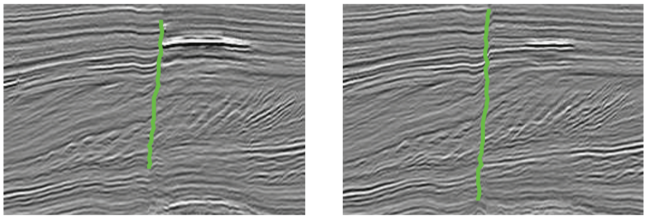
(c) Highlighted fault points

Fig. 4: Seismic section and the highlighted fault points



(a) (b) (c) (d) (e) (f) (g)

Fig. 5: (a) Detected fault features based on Hough transform. (b) Fault features after false feature removal. (c) Zig-zag fault line optimized as in Eq. (7). (d) Smoothed version of (c). (e) Fault line after combining (b) and (d). (f) Fault line detected by proposed method. (g) Fault line detected by the method in [10].



(a) Faults in Inline 246

(b) Faults in Inline 268

Fig. 6: Detected faults in the neighboring seismic sections

6. REFERENCES

[1] R. Pepper and G. Bejarano, "Advances in seismic fault interpretation automation," in *Search and Discovery Article 40170, Poster presentation at AAPG Annual Convention*, 2005, pp. 19–22.

[2] Satinder Chopra and Kurt J. Marfurt, *Seismic Attributes for Prospect Identification and Reservoir Characterization*, Society of Exploration Geophysicists and European Association of Geoscientists and Engineers, 2007.

[3] Peter P. Van Bemmel and Randolph E. F. Pepper, "Seismic signal processing method and apparatus for generating a cube of variance values," Nov. 21 2000, US Patent 6,151,555.

[4] Ahmed Adnan Aqrawi and Trond Hellem Boe, "Improved fault segmentation using a dip guided and modified 3d sobel filter," in *2011 SEG Annual Meeting*, 2011, pp. 999–1003.

[5] J. Song, X. Mu, Z. Li, C. Wang, and Sun Y., "A faults identification method using dip guided facet model edge detector," in *2012 SEG Annual Meeting*, 2012, pp. 1–5.

[6] David Gibson, Michael Spann, Jonathan Turner, and Timothy Wright, "Fault surface detection in 3-d seismic data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 9, pp. 2094–2102, 2005.

[7] Israel Cohen, Nicholas Coult, and Anthony A Vassiliou, "Detection and extraction of fault surfaces in 3d seismic data," *Geophysics*, vol. 71, no. 4, pp. P21–P27, 2006.

[8] Nasher M. AlBinHassan and Kurt J. Marfurt, "Fault detection using hough transforms," in *2003 SEG Annual Meeting*, 2003.

[9] P. Jacquemin and J. Mallet, "Automatic faults extraction using double hough transform," in *2005 SEG Annual Meeting*, 2005, pp. 755–758.

[10] D. Hale, "Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3d seismic images," *GEOPHYSICS*, vol. 78, no. 2, pp. O33–O43, 2013.

[11] Kurt J. Marfurt, R. Lynn Kirlin, Steven L. Farmer, and Michael S. Bahorich, "3-D seismic attributes using a semblance-based coherency algorithm," *Geophysics*, vol. 63, no. 4, pp. 1150–1165, 1998.

[12] Rafael C. Gonzalez and Richard E. Woods, *Digital image processing (three edition)*, Prentice Hall, 2008.

[13] Opendtect, "Netherland Offshore F3 Block Complete," `http://www.opendtect.org/index.php/share-seismic-data/osr.html`.