

pySEISPROC: A python based open-source geophysical package for visualization and fault interpretation of 2D or 3D seismic data

Ratul Kishore Saha*, Tiash Ghosh, Aurobinda Routray, Swanand Khare, IIT Kharagpur
Sanjai Kumar Singh, Rajeev Tandon, CS Bahuguna, ONGC, India
ratulkishoresaha97@gmail.com

Keywords

pySEISPROC, Faults, Structural Dip, Semblance, Eigen-Structure, Gradient-Structure Tensor, Python

Abstract

This paper aims to develop an open-source python based interactive geophysical package called pySEISPROC. pySEISPROC is an integrated package designed for 3D seismic data processing. It allows geophysicists to rapidly interact with huge datasets. It contains visualization tools for better understanding of the data and analysis tools for fault extraction in the 3D seismic volume. This study also aims to introduce standard noise suppression techniques for data preprocessing in order to increase efficiency of the prediction algorithm. Automatic fault extraction is performed using traditional deterministic approaches such as structural dip, semblance, eigenstructure, gradient structure tensor based coherence, etc.

Introduction

Seismic faults are typically imaged as discontinuities in the data. Faults provide insight into many subsurface hydrocarbon traps. Hence, a precise depiction of faults is important for reservoir characterization and exploration of the field. Earlier, faults were manually interpreted by visual recognition of discontinuities in the seismic data. But these methods consume huge time and are sensitive to interpreters' bias. To increase the accuracy of interpretation, algorithms are being developed for effective semi-automatic or automatic fault extraction for highlighting fault points followed by model-based fault tracking in 3D seismic data.

Taner et al proposed estimation of dip using complex trace analysis [1] of seismic images. In [2, 3, 4], authors proposed discontinuity analysis algorithms for extraction of faults in 3D seismic volume. These are the most frequently used seismic attributes for fault interpretation. Towards the end of 1999, the semblance based coherence algorithm was slightly modified. As the fault is dominating along the same dip, semblance was calculated by aligning the analysis window along the dip or azimuthal angle. Further,

it is seen that noise distorts the seismic image and interferes with the fault detection accuracy. In order to cater the noise of seismic data, another structural attribute named gradient structure tensor [5] was proposed for seismic data interpretation. The authors in [6,7] introduced 3D Hough transform to detect fault planes in 3D seismic volume. Later, it was extended to the double Hough transform [8]. Among all the fault enhancement algorithms, the ant tracking [9] was considered to be most effective. All these traditional approaches are data dependent. Zhang et. al first introduced machine learning based algorithm [10] for fault extraction. In 2017, Xiong et. al extended the existing machine learning algorithms to a convolutional neural network framework [11] for seismic fault detection in 3D volume.

Presently, python is widely used for seismic processing and analysis of 3D seismic data. Due to ease of interactive analysis almost all researchers use python. The purpose of this work is to develop a python based open-source geophysical package for fault interpretation in 3D seismic volume. This package provides utilities for data input, data pre-conditioning, calibration of data and other image processing techniques for automatic fault extraction. The fault points are extracted and highlighted using various traditional deterministic approaches. Denoising techniques are introduced for pre-conditioning the seismic data. The results of the 3D seismic attributes are compared on the original and filtered data is presented.

Methodology

The integrated geophysical package contains utilities for visualization and analysis of seismic data. The implementation of the proposed package relies on *numpy*, *scipy*, *segpy*, *segio*, etc. Figure 1 shows the tree diagram for the pySEISPROC package. The detailed descriptions of the pySEISPROC utilities are given in Table 1. This package provides some state-of-the-art algorithms for fault interpretation in 3D seismic volume. The backend algorithms involved for

pySEISPROC: A python based open-source geophysical package for visualization and fault interpretation of 2D or 3D seismic data

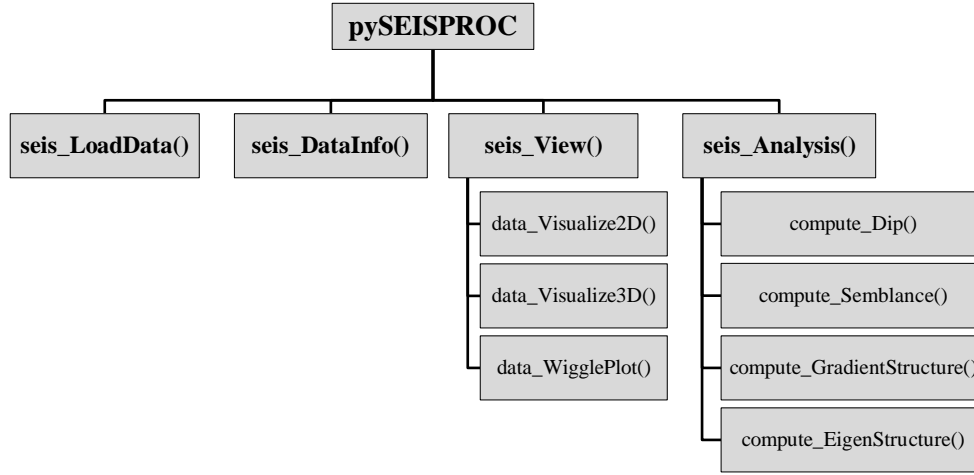


Figure 1: Tree Diagram of the python based geophysical package (PySEISProc)

fault interpretation have been discussed below.

A. Estimation of dip attribute:

Dip is a valuable attribute in seismic data interpretation as it discerns faults in a seismic section. In the developed package *compute_Dip()* the 3D structural dip of the seismic volume is estimated by complex-trace analysis[1]. In [1] the instantaneous frequency (w), instantaneous wave number k_x , k_y and k_z is defined as in Equation (1), (2), (3), (4) respectively

$$w(x, y) = \frac{\partial \phi}{\partial t} = \frac{\partial}{\partial t} ATAN(u^H, u) = \frac{u \frac{\partial u^H}{\partial t} - u^H \frac{\partial u}{\partial t}}{(u)^2 + (u^H)^2} \quad (1)$$

$$k_x(x, y, z) = \frac{\partial \phi}{\partial x} = \frac{u \frac{\partial u^H}{\partial x} - u^H \frac{\partial u}{\partial x}}{(u)^2 + (u^H)^2} \quad (2)$$

$$k_y(x, y, z) = \frac{\partial \phi}{\partial y} = \frac{u \frac{\partial u^H}{\partial y} - u^H \frac{\partial u}{\partial y}}{(u)^2 + (u^H)^2} \quad (3)$$

$$k_z(x, y, z) = \frac{\partial \phi}{\partial z} = \frac{u \frac{\partial u^H}{\partial z} - u^H \frac{\partial u}{\partial z}}{(u)^2 + (u^H)^2} \quad (4)$$

where ϕ denotes the instantaneous phase, $u(x, y, z)$ denotes the input seismic data, $u^H(x, y, z)$ denotes its Hilbert Transform with respect to time t , and $ATAN2$ denotes the arctangent function whose output varies between $-\pi$ and $+\pi$.

The dip angle (θ) is thus formulated as-

$$\theta = \tan^{-1} \sqrt{\frac{k_x^2 + k_y^2}{k_z}} \quad (5)$$

B. Discontinuity mapping:

i. Semblance based coherence:

Semblance is an extremely powerful attribute for discontinuity mapping like faults, fractures and structure analysis. In *pySEISPROC*, *compute_Semblance()* method computes semblance of 3D seismic volume. In [2, 4] the semblance is estimated along the analysis window (rectangular or elliptical) of size $2r+1$ guided by the dip scan. In [2], the semblance $D(x, z)$ is calculated as –

$$D(x, z) = \ln \left[\frac{\sum_{j=-r}^r (\sum_{i=-r}^r S(x+i, z+j))^2}{(2r+1) \sum_{i=-r}^r \sum_{j=-r}^r S(x+i, z+j)^2} \right] \quad (5)$$

where $S(x, z)$ is the intensity of the seismic signal at point (x, z) . In a seismic section, x and z represents the co-ordinates on crossline and depth direction, respectively.

ii. Gradient Structure Tensor based coherence:

Gradient-Structure Tensor (GST) [5] is another structural attribute in seismic data interpretation study. In this package, the *compute_GradientStructure()* method computes semblance of 3D seismic volume. The first step of estimating GST is to calculate the gradient $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$, $\frac{\partial u}{\partial z}$. In [5], the authors recommended calculating these derivatives by convolving the seismic data with the derivatives of a Gaussian filter (G).

$$G(x_i, y_i, z_i, \sigma) = \exp[-(x_i^2 + y_i^2 + z_i^2) / \sigma^2] \quad (6)$$

where x_i , y_i and z_i are the distances along the x , y

pySEISPROC: A python based open-source geophysical package for visualization and fault interpretation of 2D or 3D seismic data

SL No.	Method Name	Description
1.	<i>seis_LoadData()</i>	Reads 2D/3D seismic data in .sgy format.
2.	<i>seis_DataInfo()</i>	Displays inline, crossline and depth information from header.
3.	<i>seis_View</i>	2D/3D visualization of the seismic data.
	a. <i>data_Visualize2D():</i>	2D visualization of seismic data.
	b. <i>data_Visualize3D():</i>	3D visualization of seismic data.
	c. <i>data_WigglePlot():</i>	Visualize seismic data as wave-forms of the recorded data.
4.	<i>seis_Analysis</i>	Helps in analysis and fault interpretation of seismic data.
	a. <i>compute_Dip():</i>	Computes 3D structural dip of the seismic data.
	b. <i>compute_Semblance():</i>	Computes semblance based coherence in seismic volume.
	c. <i>compute_EigenStructure():</i>	Computes eigen-structure based coherence in seismic volume.
	d. <i>compute_GradientStructure():</i>	Computes gradient-structure tensor based coherence in seismic volume.

Table 1: Description of PySEISProc utilities

and z axes of the j^{th} trace from the point at which the derivative is being analyzed, and σ is the scale parameter. The second step consists of mapping the gradient to the tensor using the dyadic product and average the tensor components T at scale σ . The gradient structure tensor is defined by:

$$T = \overline{gg^T} \quad (7)$$

Eigen value of T is calculated and the coherence is estimated as-

$$C_{GST} = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \quad (8)$$

where λ_1 and λ_2 are the maximum and minimum eigen values of T .

iii. Eigen Structure based coherence attribute:

The *compute_GradientStructure()* method estimates the eigenstructure based coherence. Eigenstructure [3] is calculated on an analysis window (elliptical/square) guided along the dip angle. The covariance matrix C for each analysis window is calculated as:

$$C = D^T D \quad (9)$$

where
$$D = \begin{bmatrix} d_{11} & \dots & d_{1j} \\ \vdots & \ddots & \vdots \\ d_{N1} & \dots & d_{Nj} \end{bmatrix}$$

In this representation of the data, a single entry d_{nj} is therefore the amplitude of the n^{th} sample of the j^{th} trace. Eigen-Structure based coherence estimation uses the numerical trace of the covariance matrix(C) denoted by $Tr(C)$. The numerical trace of C may be expressed in as:

$$Tr(C) = \sum_{j=1}^j \sum_{n=1}^N d_{nj}^2 = \sum_{j=1}^j c_{jj} = \sum_{j=1}^j \lambda_j \quad (10)$$

EigenStructure based coherence estimate (E_c) is formulated as-

$$E_c = \frac{\lambda_1}{Tr(C)} = \frac{\lambda_1}{\sum_{j=1}^j c_{jj}} = \frac{\lambda_1}{\sum_{j=1}^j \lambda_j} \quad (11)$$

where λ_1 is the largest eigenvalue of λ_j . Equation (11) defines the eigenstructure coherence as a ratio of the dominant eigenvalue λ_1 to the total energy within the analysis cube.

pySEISPROC: A python based open-source geophysical package for visualization and fault interpretation of 2D or 3D seismic data

Experimental Results

A. Database description

The input data chosen for this experimental study is the 3D pre-stack time migrated (PSTM) data volume, located in the offshore KG Basin. The dataset contains 2535 inline sections (1982 - 4517) and 2601 cross-line sections (8800- 14000, step 2). The data is recorded for 4 seconds. In the given data, inline, crosslines, Xcdp, Ycdp starts from 189, 193, 181, 185 byte location respectively. The major faults are concentrated in the bounded region in Fig 3. which is considered for the further experimental purpose.

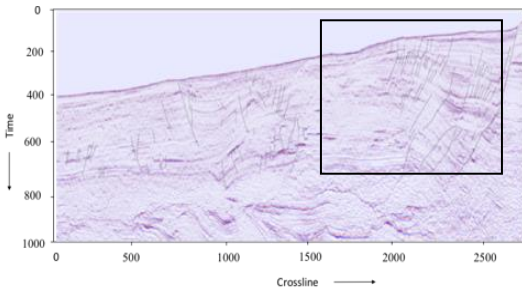


Figure 3: Manually interpreted faults

B. Visualization of 2D/3D seismic data

3D view of seismic dataset can be visualized using the `data_Visualize3D()` method and 2D view can be visualized using `data_Visualize2D()`.

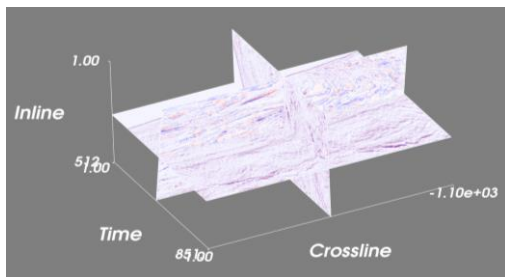


Figure 4: 3D visualization of loaded seismic volume

In this package, `data_WigglePlot()` method is incorporated to visualize the wiggle plot of specified inline section. Figure 5 shows the wiggle plot of inline section 3000 of KG Basin data.

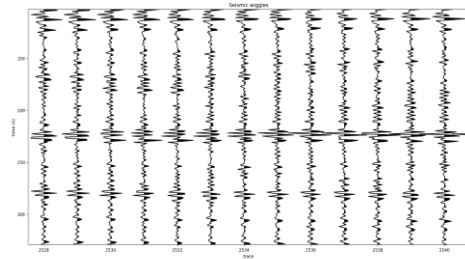


Figure 5: Wiggle plot of 2D seismic section

C. Fault interpretation in seismic volume

Seismic fault interpretation is accomplished using four algorithms as stated above. The results of all algorithms are discussed here.

I. Result of Dip attribute for fault interpretation

3D structural dip using complex trace analysis is estimated using `compute_Dip()` method. The dip angle is varied in between -90 to +90 degrees. There is a provision to visualize the specific range of dip angle result according to user demand.

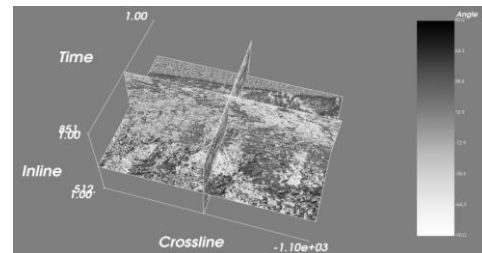


Figure 6: 3D visualization of structural dip

II. Result of discontinuity mapping using semblance attribute:

Discontinuity mapping using semblance is computed using `compute_Semblance()` method. The size of the analysis window is varied. Almost all major faults are highlighted using this attribute as compared with the bounded region in Fig 3. Figure 7 shows the 3D semblance based coherency result.

III. Result of discontinuity mapping using gradient structure tensor based coherence:

Discontinuity mapping can also be computed using the `compute_GradientStructure()` method.

pySEISPROC: A python based open-source geophysical package for visualization and fault interpretation of 2D or 3D seismic data

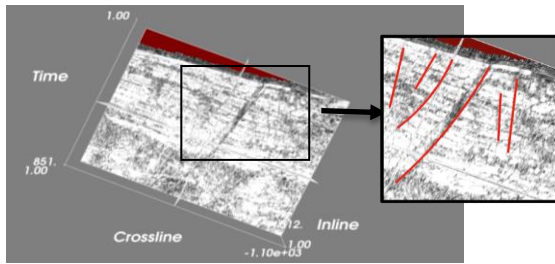


Figure 7: Semblance based discontinuity map using 5X5 analysis window

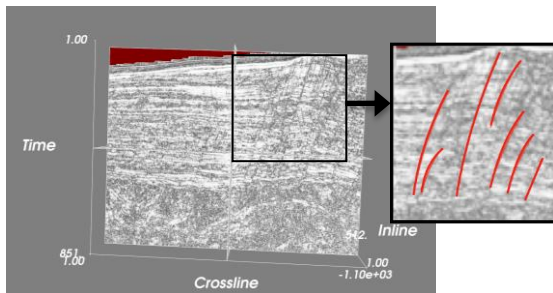


Figure 8: 3D visualization of GST based coherence ($\square=1$)

The parameter \square is fixed at different values. Figure 8 shows 3D gradient structure based coherence.

From the above discussion it may be concluded that semblance based coherence gives promising result for discontinuity mapping. So semblance attribute is chosen for the next phase of the experiment. As the above mentioned seismic attributes is data dependent, erroneous result in discontinuity mapping may arise due to noise. Hence, pySEISPROC involves pre-processing of seismic data for noise removal. Noise suppressing filters are applied on each seismic trace with different cut off frequencies. Figure 9 shows the noise reduction due to application of butter-worth filter with cut-off frequency of 60 Hz. After filtering the attributes are calculated. It is evident from Fig 10, that applying noise suppression technique improves the fault interpretation efficiency. The usage of the methods in the proposed geophysical package has been described in Fig 11.

Conclusion

Traditional deterministic approaches have been incorporated in this package for fault detection. By varying the cut off frequency of the denoising filter, high frequency noises are eliminated. Noise removal

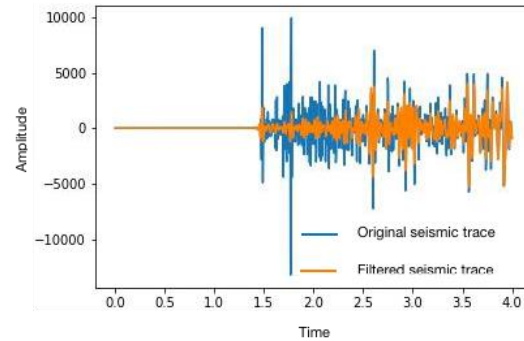


Figure 9: Noise reduction due to low-pass filtering

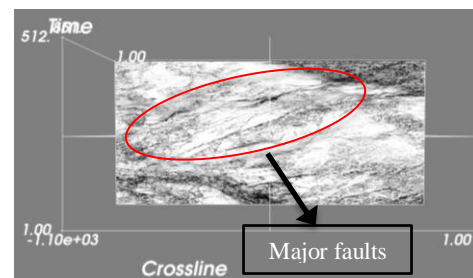


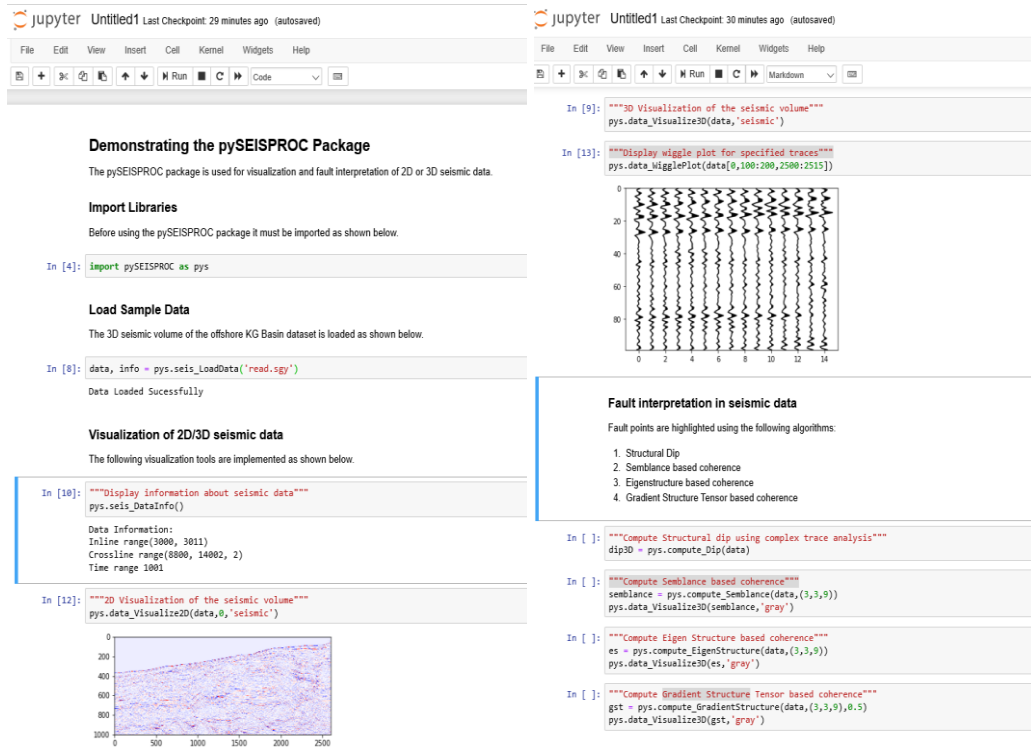
Figure 10: Time slice view of highlighted fault points in filtered seismic volume

results in increased efficiency of the fault extraction algorithms. All these algorithms have been implemented and tested on 3D seismic data (offshore KG Basin dataset). One of the primary goals of this study is to develop an open-source the package that allows users to incorporate customized functions of their own usage. Further, robust denoising algorithms may be adopted to obtain better results. Also, efficient deep neural network frameworks may be implemented in pySEISPROC for fast and automatic fault extraction in 3D seismic volume.

Acknowledgement

This work was undertaken by Indian Institute of Technology, Kharagpur in collaboration with Geo data Processing & Interpretation Centre (GEOPIC), ONGC, Dehradun. under the aegis of ONGC-PANIIT projects.

pySEISPROC: A python based open-source geophysical package for visualization and fault interpretation of 2D or 3D seismic data



Demonstrating the pySEISPROC Package
The pySEISPROC package is used for visualization and fault interpretation of 2D or 3D seismic data.

Import Libraries
Before using the pySEISPROC package it must be imported as shown below.

```
In [4]: import pySEISPROC as pys
```

Load Sample Data
The 3D seismic volume of the offshore KG Basin dataset is loaded as shown below.

```
In [8]: data, info = pys.seis_LoadData('read.sgy')
```

Data Loaded Successfully

Visualization of 2D/3D seismic data
The following visualization tools are implemented as shown below.

```
In [10]: """Display information about seismic data"""
pys.seis_DataInfo()

Data Information:
Inline range(3000, 3011)
Crossline range(8800, 14002, 2)
Time range 1001
```

```
In [12]: """2D Visualization of the seismic volume"""
pys.data_Visualize2D(data,0,'seismic')
```

3D Visualization of the seismic volume

```
In [9]: """3D Visualization of the seismic volume"""
pys.data_Visualize3D(data,'seismic')
```

Display wiggle plot for specified traces

```
In [13]: """Display wiggle plot for specified traces"""
pys.data_WigglePlot(data[0,100:200,2500:2515])
```

Fault interpretation in seismic data
Fault points are highlighted using the following algorithms:

1. Structural Dip
2. Semblance based coherence
3. Eigenstructure based coherence
4. Gradient Structure Tensor based coherence

```
In [ ]: """Compute Structural dip using complex trace analysis"""
dip3D = pys.compute_Dip(data)
```

```
In [ ]: """Compute Semblance based coherence"""
semblance = pys.compute_Semblance(data,(3,3,9))
pys.data_Visualize3D(semblance,'gray')
```

```
In [ ]: """Compute Eigen Structure based coherence"""
es = pys.compute_EigenStructure(data,(3,3,9))
pys.data_Visualize3D(es,'gray')
```

```
In [ ]: """Compute Gradient Structure Tensor based coherence"""
gst = pys.compute_GradientStructure(data,(3,3,9),0.5)
pys.data_Visualize3D(gst,'gray')
```

Figure 11: Code snippet for usage of pySEISPROC package

References

- [1] Taner, M. Turhan, Fulton Koehler, and R. E. Sheriff. "Complex seismic trace analysis." *Geophysics* 44, no. 6 (1979): 1041-1063.
- [2] Marfurt, Kurt J., R. Lynn Kirlin, Steven L. Farmer, and Michael S. Bahorich. "3-D seismic attributes using a semblance-based coherency algorithm." *Geophysics* 63, no. 4 (1998): 1150-1165.
- [3] Gersztenkorn, Adam, and Kurt J. Marfurt. "Eigen structure-based coherence computations as an aid to 3-D structural and stratigraphic mapping." *Geophysics* 64, no. 5 (1999): 1468-1479.
- [4] Marfurt, Kurt J., V. Sudhaker, Adam Gersztenkorn, Kelly D. Crawford, and Susan E. Nissen. "Coherency calculations in the presence of structural dip." *Geophysics* 64, no. 1 (1999): 104-111.
- [5] Bakker, P., 2003, Image structure analysis for seismic interpretation: Ph.D. thesis, Technische Universiteit Delft.
- [6] Wang, Zhen, and Ghassan AlRegib. "Interactive fault extraction in 3-D seismic data using the Hough transform and tracking vectors." *IEEE Transactions on Computational Imaging* 3, no. 1 (2016): 99-109.

- [7] Wang, Zhen, and Ghassan AlRegib. "Fault detection in seismic datasets using Hough transform." In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2372-2376. IEEE, 2014.
- [8] Jacquemin, Pierre, and Jean-Laurent Mallet. "Automatic faults extraction using double hough transform." In *SEG Technical Program Expanded Abstracts 2005*, pp. 755-758. Society of Exploration Geophysicists, 2005.
- [9] Pedersen, Stein Inge, Trygve Randen, Lars Sonneland, and Øyvind Steen. "Automatic fault extraction using artificial ants." In *SEG Technical Program Expanded Abstracts 2002*, pp. 512-515. Society of Exploration Geophysicists, 2002.
- [10] Zhang, Chiyuan, Charlie Frogner, M. Araya-Polo, and D. Hohl. "Machine-learning based automated fault detection in seismic traces." In *76th EAGE Conference and Exhibition 2014*. 2014.
- [11] Xiong, Wei, Xu Ji, Yue Ma, Yuxiang Wang, Nasher M. AlBinHassan, Mustafa N. Ali, and Yi Luo. "Seismic fault detection with convolutional neural network." *Geophysics* 83, no. 5 (2018): O97-O103.