

# AUTOMATIC FAULT LOCALIZATION USING THE GENERALIZED EARTH MOVER’S DISTANCE

Ludwig Schmidt, Chinmay Hegde, Piotr Indyk

Jonathan Kane, Ligang Lu, Detlef Hohl

Massachusetts Institute of Technology  
Cambridge, MA 02139

Shell International E&P, Inc.  
Houston, TX 77082

## ABSTRACT

Localizing fault lines and surfaces in seismic subsurface images is a daunting challenge. Existing state-of-the-art approaches usually involve visual interpretation by an expert, but this is time-consuming, expensive and error-prone. In this paper, we propose some initial steps towards a new algorithmic framework for automatic fault localization. The core of our approach is a deterministic model for 2D images that we call the *Constrained Generalized Earth Mover’s Distance* (CGEMD) model. We propose an algorithm that returns the best approximation in the model for any given input 2D image  $X$ ; the output of this algorithm is then post-processed to reveal the locations of the faults in the image. We demonstrate the validity of this approach on a number of synthetic and real-world examples.

**Index Terms**— Sparse recovery, Earth Mover’s Distance, seismic signal processing, min-cost flows.

## 1. INTRODUCTION

A key goal in seismic data processing and interpretation is the detection of geological features in images. A particularly interesting class are *faults*, or shear-like discontinuities. Analysis of subsurface faults can convey a wealth of spatio-temporal information about geological events. Therefore, there is a strong practical need to accurately localize faults in a seismic image.

However, fault localization can be a daunting challenge. Seismic images are typically contaminated with noise, complicating the detection of even strong discontinuities. Fault lines (and surfaces) are also of varied shapes and sizes, and can themselves exhibit unpredictable discontinuities. One approach to fault localization is to perform visual inspection and interpretation by an expert. However, this procedure can be time-consuming, expensive, subjective, and error-prone; see Section 2.1 for a brief review of some other existing approaches.

In this paper, we propose some initial steps towards a new algorithmic framework for *automatic* fault localization in seismic images. The core of our framework is a deterministic model for 2D images, called the *Constrained Generalized Earth Mover’s Distance* (CGEMD) model. A similar signal model was recently proposed and analyzed in [1, 2]. The definition of this model consists of two main ingredients. The first ingredient is *sparsity*, a notion that has gained significant traction in signal processing applications over the last few years. In the seismic imaging context, we leverage this notion to model the fact that a subsurface image is likely to consist of only a *small number of strong reflectors*. The second ingredient is the classical Earth Mover’s Distance (EMD), a metric commonly used to compare probability distributions. We leverage this notion to model the fact that the profiles of subsurface reflectors are mostly *continuous*, except at the locations of the faults. Our proposed model is

flexible enough to include 2D seismic images with even non-straight (“wiggly”) and broken fault lines.

Our overall algorithm proceeds in two stages. First, given an arbitrary image  $X$ , we find a near-optimal *approximation* of  $X$  that does indeed lie in the CGEMD model. The algorithm proposed in this paper is a generalized (and computationally more efficient) version of our previous algorithm [1]. It is also supported by rigorous stability guarantees. We show that the model projection can be discovered by an iterative algorithm that solves a small number of *min-cost flow* problems over a specially defined graph. Each intermediate problem can be solved using existing tools from graph optimization, and therefore the overall model approximation of  $X$  can be obtained efficiently in near-linear time. Second, we post-process the support (i.e., the locations of the nonzeros) in the derived model approximation of  $X$  to robustly localize the faults. This can again be done in time linear in the size of the support. An easy modification of this post-processing step enables an extension of this approach to 3D seismic images.

We demonstrate, via numerical experiments, the advantages of the CGEMD model and the associated approximation algorithm in the context of automatic fault detection. We test our approach on a number of example images, both in the presence and absence of heavy amounts of noise. Our simulations suggest that our proposed approach succeeds even for images with a SNR equal to -5 dB. A synthetic test example is displayed in Fig. 1; see Section 4 for several more synthetic and real-world test images.

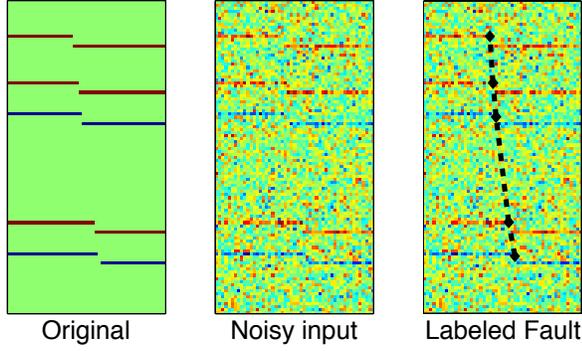
## 2. BACKGROUND

### 2.1. Related work in seismic fault detection

Considerable effort has been invested into discovering automatic methods for fault surface extraction; see, for example, the recent article [3] and references therein. Some example approaches include texture classification approaches [4], coherence-based methods [5], filtering-based methods [6], and greedy optimization algorithms to extract dominant paths in seismic images [7]. Most of these methods do not seem to work well in the presence of high levels of noise.

To mitigate this, one option is to run a traditional *denoising* procedure on the seismic images, prior to fault extraction. Traditional image denoising approaches (e.g. PDE-based image smoothing) tend to suppress fault-like features; therefore, special fault-preserving diffusion algorithms have been proposed [8].

An alternate class of approaches possesses a semi-automatic flavor: the available data is pre-processed using any of a number of techniques, and the processed data is presented to a human expert using an advanced visualization method. The expert then manually labels the faults in the image [9]. Semi-automatic methods seem



**Fig. 1.** Automatic fault localization in a seismic image. (left) We construct an original (test) image comprising a small number of sub-surface layers that are subjected to a single linear shear-like fault at a particular angle. (center) We contaminate this image with a high amount of noise, equivalent to a SNR of -5 dB, and present the noisy image as input to our fault localization algorithm. (right) Despite the high level of noise, our algorithm is able to locate the fault as indicated by the black dotted line.

to be the prevalent approach in practice; however, they are time-consuming, expensive, subjective, and error-prone. Below, we describe a novel fully automatic computational approach for fault extraction that can potentially be integrated into larger semi-automatic approaches.

## 2.2. Structured Sparsity

A key ingredient in our proposed signal model is the notion of *sparsity*. A signal  $x \in \mathbb{R}^n$  is said to be  $k$ -sparse in an orthonormal basis  $\Psi$  if at most  $k \leq n$  coefficients of the basis expansion  $\alpha = \Psi^T x$  are nonzero. In this paper, we assume that the basis  $\Psi$  is the identity matrix. The *support* of  $x$  (denoted by  $\text{supp}(x) \subseteq [n]$ ) is defined as the set of indices corresponding to the nonzero entries of  $x$ . This can equivalently be represented by a binary vector,  $s(x) \in \{0, 1\}^n$ , with at most  $k$  ones.

In many practical situations, it is reasonable to assume the availability of additional information about the *structure* of the nonzeros in a  $k$ -sparse signal  $x$ . As an example, consider the class of one-dimensional piecewise smooth signals. It is well known that such signals are sparse in the wavelet basis; moreover, the wavelet coefficients lie approximately on a rooted connected tree [10]. As another example, consider a push-to-talk communications signal where the information transmission is “bursty”, i.e., transmission happens infrequently and in short windows. Such signals are sparse in the identity basis; moreover, the nonzeros occur in *blocks*. Refer to the survey [11] for several more examples of structured sparsity models.

In all above cases, we are interested in  $k$ -sparse signals  $x$  with only a small number of permitted configurations of the support  $\text{supp}(x)$ . We call such a set of signals a *model*  $\mathcal{M}_k$  and formally define it as a set of allowed supports  $\mathbb{M}_k = \{\Omega_1, \Omega_2, \dots, \Omega_{a_k}\}$ . As in the general  $k$ -sparse case, given a signal  $x$ , we often seek a support  $\Omega$  such that  $\Omega \in \mathbb{M}_k$  (hence  $x_\Omega \in \mathcal{M}_k$ ), and  $\|x - x_\Omega\|_2$  is minimized. We can analogously define a *model projection* algorithm as a procedure  $M(x, k)$  which returns the support of the best  $k$ -term approximation of a given signal under the model  $\mathcal{M}_k$ , i.e.,  $\Omega = M(x, k)$ .

## 3. FAULT LOCALIZATION FRAMEWORK

Below, we interpret the signal  $x \in \mathbb{R}^n$  as a matrix  $X \in \mathbb{R}^{h \times w}$  with  $n = hw$ . For such a matrix, we define  $\text{supp}(X) \subseteq [h] \times [w]$  as the set of nonzero entries accordingly. Given the support of a matrix  $\Omega$ , we define the support of a column as  $\text{col-supp}(\Omega, c) = \{r \mid (r, c) \in \Omega\}$ .

### 3.1. Signal Model

The core ingredient of our signal model is the classical *Earth Mover’s Distance* (EMD), commonly used as a metric between probability distributions and successfully applied to several problems in computer vision and image retrieval [12]. We define a generalized notion of the EMD that will be suitable for our purposes.

**Definition 1** (Generalized EMD). *The Generalized EMD (GEMD) of two index sets  $P, Q \subseteq [h]$  with  $|P| = |Q| = s$  is defined as:*

$$\text{GEMD}_f(P, Q) = \min_{\pi: P \rightarrow Q} \sum_{p \in P} f(|p - \pi(p)|),$$

where  $\pi$  ranges over all one-to-one mappings from  $P$  to  $Q$  and  $f$  is a function from  $\{0, 1, \dots, h\}$  to  $\mathbb{R}$ .

**Definition 2** (Generalized EMD of a matrix support). *Let  $\Omega \subseteq [h] \times [w]$  be the support of a matrix with exactly  $s$ -sparse columns, i.e.,  $|\text{col-supp}(\Omega, c)| = s$  for  $c \in [w]$ . Then the GEMD of  $\Omega$  is defined as*

$$\text{GEMD}_f(\Omega) = \sum_{c=1}^{w-1} \text{GEMD}_f(\text{col-supp}(\Omega, c), \text{col-supp}(\Omega, c+1)),$$

where  $f$  is again a function from  $[s]$  to  $\mathbb{R}$ .

**Definition 3** (Maximum support distance). *Let  $P$  and  $Q$  be two index sets with  $|P| = |Q| = s$ . Moreover, let  $(p_1, \dots, p_s)$  and  $(q_1, \dots, q_s)$  be the elements of  $P$  and  $Q$  in ascending order. Then the maximum support distance of  $P$  and  $Q$  is*

$$\text{MSD}(P, Q) = \max_{i \in [s]} |p_i - q_i|.$$

As before, we extend this definition to the support of a matrix with exactly  $s$ -sparse columns,  $\Omega \subseteq [h] \times [w]$ :

$$\text{MSD}(\Omega) = \sum_{c=1}^{w-1} \text{MSD}(\text{col-supp}(\Omega, c), \text{col-supp}(\Omega, c+1)).$$

**Definition 4** (Constrained Generalized EMD model). *The Constrained Generalized EMD model (CGEMD) is the structured sparsity model  $\mathcal{M}_{k,B,f,\Delta}$  defined by the set of supports*

$$\mathbb{M}_{k,B,f,\Delta} = \{\Omega \subseteq [h] \times [w] \mid \text{GEMD}_f(\Omega) \leq B \text{ and } \text{MSD}(\Omega) \leq \Delta \text{ and } |\text{col-supp}(\Omega, c)| = s \text{ for } c \in [w]\}.$$

The CGEMD model is a generalization of the Constrained EMD model introduced and analyzed in [1, 2]. For  $f(x) = x$  and  $\Delta = h$ , the CGEMD model is identical to the EMD model. We refer the reader to these two papers for a more detailed discussion of the EMD model. Here, we focus on the two main modifications.

We introduce the maximum support distance in order to decrease the computational complexity of our model-approximation algorithm. The algorithm is based on a reduction to the min-cost

flow problem (see section 3.2) and the size of the underlying graph is  $O(wh^2)$  for the CEMD model. For the CGEMD model, we can run the same algorithm on a graph of size  $O(wh\Delta)$ . Since the signals occurring in some applications like fault detection usually have only small variations in the support locations, we can choose a value  $\Delta \ll h$  and thereby significantly increase the performance of the model-approximation algorithm.

Associating an arbitrary cost with the standard EMD-distance via the cost function  $f$  allows us to more carefully model the underlying problem. In the context of fault detection, we can adjust the costs so that small variations in the support become cheaper, therefore making the subsequent fault detection step more robust to small variations in the fault lines.

### 3.2. Algorithm

Our overall technique works in two stages. In the first stage, given an arbitrary seismic image, we perform a *model-approximation* step in order to extract the best  $s$  paths from left to right in the image, corresponding to the dominant subsurface layers. In the second stage, we post-process the  $s$  paths to localize the faults. For clarity, we present the two stages as separate algorithms.

#### 3.2.1. Model Approximation

In order to find a good projection of an arbitrary signal into the CGEMD model, we use the characterization of the EMD as a min-cost matching and embed the approximation problem into a sequence of min-cost flow instances. The basic reduction is described in [1]. We need the following modifications to the flow network:

- Instead of fully connected layers, we only connect each node to the nodes in the next layer with support distance at most  $\Delta$ . Hence the out-degree of each node is at most  $2\Delta + 1$ .
- We modify the edge costs to reflect the GEMD: the cost of an edge from  $v_{i,j}$  to  $v_{k,j+1}$  is  $\lambda f(|i - k|)$ .

Figure 2 illustrates the above construction with an example. Since solving a min-cost flow problem on  $G_{X,k,f,\Delta,\lambda}$  still corresponds to the optimization problem  $\max_{\Omega \in \mathbb{M}_{k,n,k,f,\Delta}} \|X_{\Omega}\|_2^2 + \lambda \text{GEMD}_f(\Omega)$ , we can perform a binary search over  $\lambda$  to find a suitable trade-off between the signal approximation and GEMD terms. We refer the reader to [1, 2] for the full model-projection algorithm and corresponding correctness proofs.

For solving the embedded min-cost flow problems, we use a capacity scaling algorithm because the per-column sparsity  $s$ , which is also the capacity at the source in the network, is typically small. Each scaling step can be implemented with a shortest path search, resulting in a total running time of  $O(tsn\Delta \log n)$ , where  $t$  is the number of calls to `MINCOSTFLOW`. We use a stopping criterion based on the number of iterations  $t$  in order to get a controllable running time.

---

#### Algorithm 1 Fault extraction algorithm

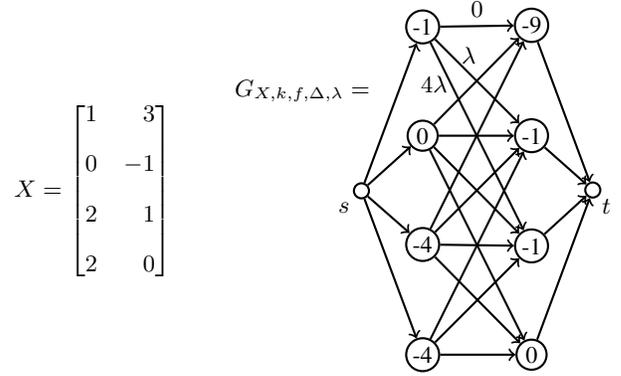
---

```

function FAULTEXTRACT( $\Omega, \alpha$ )
   $L \leftarrow \emptyset$ 
  for  $i = 1$  to  $k$  do
     $p_i \leftarrow i^{\text{th}}$  path in  $\Omega$ 
     $m \leftarrow$  median  $y$ -displacement in  $p_i$ 
    for  $X_j \in p_i$  do
      if  $y$ -displacement in  $X_j > m + \alpha$  then
         $L \leftarrow L \cup X_j$ 
  Return  $L$ 

```

---



**Fig. 2.** A signal  $X$  with the corresponding flow network  $G_{X,k,f,\Delta,\lambda}$  for  $f(x) = x^2$  and  $\Delta = 2$ . The node costs are the negated, squared coefficients of the corresponding signal components. All capacities in the flow network are 1. The edge costs are given by  $\lambda f(x)$ , where  $x$  is the vertical distance between the start and end node of an edge (for instance, the edge cost  $4\lambda$  comes from  $2^2\lambda$  because the edge connects two nodes with vertical distance 2). Most edge costs are omitted for clarity. Note that due to the choice of  $\Delta = 2$ , there is no edge between the top left and bottom right vertices (and also the bottom left and top right vertices).

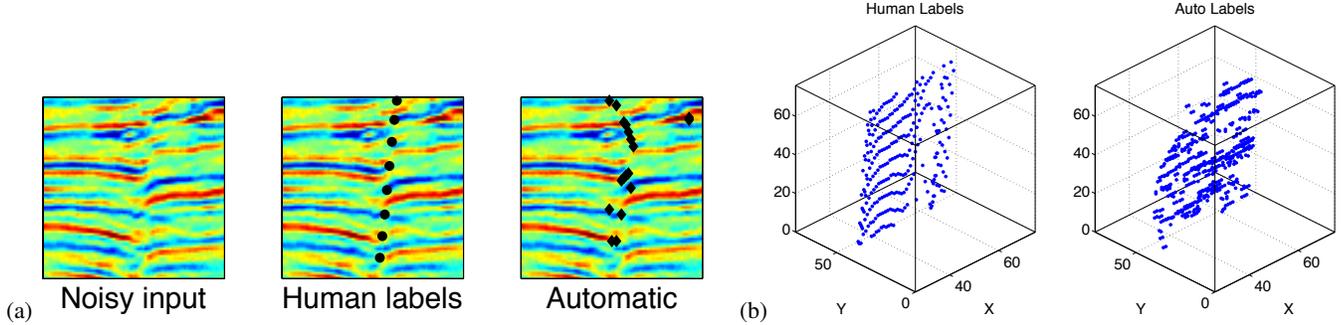
#### 3.2.2. Fault Extraction

The model-projection algorithm returns a binary matrix  $\Omega \in \{0, 1\}^{h \times w}$  that comprises the support of the (near)-optimal approximation of the input image  $X$  in the CGEMD model. Essentially, the matrix  $\Omega$  consists of  $k$  paths from left to right, with a global spatial displacement no greater than  $B$ . Furthermore, the local (column-wise) displacement of each path is also limited by the maximum support distance  $\Delta$ .

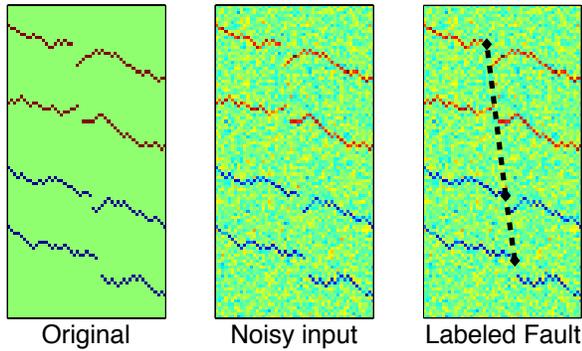
For simplicity, consider the path traced by any one of these  $k$  paths. This path (typically) would coincide with one of the  $k$  dominant reflectors in the subsurface. Therefore, this path is expected to be mostly continuous, *except* where the subsurface reflector intersects with the fault profile. At these precise locations, the path will experience a jump in the vertical direction by several pixels. If the fault profile is significantly large, such a vertical jump can be observed at paths corresponding to several flows.

This intuition motivates an algorithm to identify the fault locations from the optimal support. The algorithm simply returns a list of 2D image locations where the vertical jump of the corresponding flow path differs from the median vertical displacement of all the flows combined by at least a scalar parameter  $\alpha$ . The optimal choice of  $\alpha$  is a function of the maximum vertical displacement exacted by the fault (sometimes referred to in the literature as the *fault throw*) as well as the median incline of each of the subsurface reflectors, and will vary from dataset to dataset.

Once the complete list of potential fault locations is identified, the actual fault profiles are reconstructed using standard interpolation techniques. The simplest option is to perform piecewise linear interpolation between the labeled locations, but more sophisticated interpolation schemes are also possible. For instance, prior to the interpolation, a simple outlier rejection can be achieved by removing spurious 2D locations that are spatially isolated from all other identified locations. This (optional) step offers some amount of robustness in the final solution. The entire procedure is summarized as Alg. 1.



**Fig. 4.** Results of our fault localization algorithm on a real-world seismic dataset. (a) Results on a 2D section of the Forcados-Yokri dataset [13]. Manual labeling of the fault present in this section tends to be error-prone. Instead, our algorithm is able to approximate the fault well. (b) Results on a 3D section of the same dataset. Applying our algorithm frame-by-frame yields favorable results.



**Fig. 3.** Automatic fault localization in a seismic image, challenging case. (left) We construct an original (test) image comprising a small number of nonlinear subsurface layers (strata) that are subjected to a single linear shear-like fault at a particular angle. (center) we contaminate this image with a moderate amount of noise, equivalent to an  $\text{SNR} = 0\text{dB}$ , and present the noisy image as input to our fault localization algorithm. (right) Our algorithm is able to approximate the location of the fault.

#### 4. NUMERICAL EXPERIMENTS

We test our CGEMD-based fault localization approach on some representative synthetic and real-world seismic images. Our algorithm is computationally very efficient; specifically, for each of the test cases below, the algorithm took only a few seconds to execute in MATLAB.

Figure 1 (in Section 1) illustrates a challenging test case. The test image of size  $50 \times 25$  consists of  $k = 5$  strata and a linear sloping fault. However, in this case the input to the algorithm is corrupted with a heavy amount of Gaussian noise (the  $\text{SNR}$  in this case equals  $-5\text{dB}$ ). Due to the high noise, it is very difficult to localize the fault, even via visual inspection (or “eyeballing”) of the image. Instead, we run our model-projection algorithm on the noisy image with parameters  $k = 5$  and  $B = 50$ ; identify the locations at which the flow jumps by a particular threshold parameter  $\alpha$  (equal to  $0.5\Delta$ ); and interpolate these locations via straight line segments. These line segments are overlaid onto the input image for visualization purposes. We observe that our proposed approach is able to successfully localize the fault.

Figure 3 demonstrates that our algorithm succeeds even in some-

what more complicated situations where the sparse subsurface layers are nonlinear and discontinuous, and the overall image is once again contaminated with noise (the  $\text{SNR}$  in this case equals  $0\text{dB}$ ). We use parameters  $k = 4$  and  $B = 200$  for this particular example. Since our signal model only constrains the overall cumulative EMD, our approximation algorithm is robust to wiggles in the strata and/or faults and approximates the true location of the fault well.

Figure 4 demonstrates the performance of our algorithm on a real-world data set. The Forcados-Yokri data set consists of a 3D seismic data cube that has been manually annotated at the locations of the significant faults [13]. As our first test case, we use a simple  $75 \times 75$  2D section of this dataset that contains a single fault. First, we run the model-projection algorithm with parameters  $k = 15$ ,  $B = 150$ , and locate the points where there are significant jumps in the flows. Next, in order to eliminate spurious outliers, we discard all isolated identified fault locations, i.e., identified locations that are not in close proximity with any other identified location. The remaining locations are overlaid onto the input image for visualization purposes. From Fig. 4(a), we observe that our overall method is able to successfully reconstruct the curved structure of the fault. In fact, in this particular case the manual labeler seems to have missed the curve near the top of the image.

Finally, Fig. 4(b) displays the result of our algorithm on a  $75 \times 75 \times 50$  3D section of this dataset that contains a (roughly) planar fault. To obtain this result, we simply run the 2D version of our overall algorithm frame-by-frame, and stack the results to obtain an initial estimate of the 3D fault profile. Next, we run a simple 3D-median filtering algorithm to enforce consistency, as well as reject outliers, across estimates obtained in successive frames. As visually evident from Fig. 4(b), the final result well-approximates the locations of the faults identified by the manual process. It is likely that sophisticated methods of outlier rejection and/or robust interpolation can yield even better results, but we do not pursue this direction here.

Several avenues for improvement remain. The performance of the algorithm (crucially) depends on the input sparsity parameter ( $k$ ) and the EMD-budget parameter ( $B$ ), and a fully automatic method to deduce these parameters would be of considerable interest. Also, our algorithm is suitable for seismic images with a small number of subsurface layers, but an interesting question remains: what happens when the images are not sparse? Finally, given a series of collocated sections automatically marked with faults via our algorithm, how can we reconcile different estimates in order to produce geologically consistent identifications of the fault locations? We defer such challenging questions to future work.

## 5. REFERENCES

- [1] L. Schmidt, C. Hegde, and P. Indyk, "The constrained earth mover distance model, with applications to compressive sensing," in *10th Intl. Conf. on Sampling Theory and Appl. (SAMPTA)*, 2013.
- [2] C. Hegde, P. Indyk, and L. Schmidt, "Approximation-tolerant model-based compressive sensing," in *ACM Symp. Discrete Algorithms*, Jan. 2014.
- [3] 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.
- [4] T. Randen, E. Monsen, C. Signer, A. Abrahamsen, J. Hansen, T. Sæter, and J. Schlaf, "Three-dimensional texture attributes for seismic data analysis," in *SEG Annual Meeting*, 2000.
- [5] K. Marfurt, R. Kirlin, S. Farmer, and M. Bahorich, "3D seismic attributes using a semblance-based coherency algorithm," *Geophysics*, vol. 63, no. 4, pp. 1150–1165, 1998.
- [6] I. Cohen, N. Coult, and A. Vassiliou, "Detection and extraction of fault surfaces in 3D seismic data," *Geophysics*, vol. 71, no. 4, pp. P21–P27, 2006.
- [7] S. Pederson, T. Skov, T. Randen, and L. Sønneland, "Automatic fault extraction using artificial ants," *Math. Methods and Modeling in Hydrocarbon Explor.*, vol. 7, pp. 107–116, 2005.
- [8] O. Laviaille, S. Pop, C. Germain, M. Donias, S. Guillon, N. Kesker, and Y. Berthoumieu, "Seismic fault preserving diffusion," *J. Appl. Geophysics*, vol. 61, no. 2, pp. 132–141, 2007.
- [9] K. Tingdahl and M. De Rooij, "Semi-automatic detection of faults in 3D seismic data," *Geophysical Prospecting*, vol. 53, no. 4, pp. 533–542, July 2005.
- [10] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA, 1999.
- [11] M. Duarte and Y. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Trans. Sig. Proc.*, vol. 59, no. 9, pp. 4053–4085, 2011.
- [12] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a metric for image retrieval," *Intl. J. Comp. Vision*, vol. 40, no. 2, pp. 99–121, 2000.
- [13] S. Back, C. Höcker, M. Brundiers, and P. Kukla, "Three-dimensional-seismic coherency signature of Niger Delta growth faults: integrating sedimentology and tectonics," *Basin Research*, vol. 18, pp. 323–337, 2006.