

Building realistic structure models to train convolutional neural networks for seismic structural interpretation

Xinming Wu¹, Zhicheng Geng², Yunzhi Shi², Nam Pham², Sergey Fomel², and Guillaume Caumon³

ABSTRACT

Seismic structural interpretation involves highlighting and extracting faults and horizons that are apparent as geometric features in a seismic image. Although seismic image processing methods have been proposed to automate fault and horizon interpretation, each of which today still requires significant human effort. We improve automatic structural interpretation in seismic images by using convolutional neural networks (CNNs) that recently have shown excellent performances in detecting and extracting useful image features and objects. The main limitation of applying CNNs in seismic interpretation is the preparation of many training data sets and especially the corresponding geologic labels. Manually labeling geologic features in a seismic image is highly time-consuming and subjective, which often results in incompletely or inaccurately labeled training

images. To solve this problem, we have developed a workflow to automatically build diverse structure models with realistic folding and faulting features. In this workflow, with some assumptions about typical folding and faulting patterns, we simulate structural features in a 3D model by using a set of parameters. By randomly choosing the parameters from some predefined ranges, we are able to automatically generate numerous structure models with realistic and diverse structural features. Based on these structure models with known structural information, we further automatically create numerous synthetic seismic images and the corresponding ground truth of structural labels to train CNNs for structural interpretation in field seismic images. Accurate results of structural interpretation in multiple field seismic images indicate that our workflow simulates realistic and generalized structure models from which the CNNs effectively learn to recognize real structures in field images.

INTRODUCTION

Faults and horizons are the most important structural mappings that can be extracted from a 3D seismic image. Seismic fault interpretation is important for hydrocarbon reservoir characterization (e.g., Knipe et al., 1998), well planning (Ellevset et al., 1998; Rivænas et al., 2005), building structure models (Caumon et al., 2009), and tectonic analysis (e.g., Kusznir and Karner, 2007; Baudon and Cartwright, 2008). Seismic horizon interpretation is important for stratigraphic analysis (Vail et al., 1977) and analyzing ancient depositional environment and geomorphic features (Posamentier et al., 2007).

In a seismic image, interpreters laterally track geologically consistent reflections to extract horizons while looking for reflection discontinuities to pick faults. Manually interpreting each of the horizon and fault mappings is highly labor intensive and tedious, especially in 3D large seismic images. Numerous conventional image processing techniques have been proposed to automate the structural interpretation to some extent. because faults are typically apparent as reflection discontinuities, some seismic attributes such as semblance (Marfurt et al., 1998; Hale, 2009), coherency (Marfurt et al., 1999; Qi et al., 2017; Wu, 2017), and fault likelihood (Hale, 2013; Wu and Hale, 2016) are proposed to detect faults by high-

Manuscript received by the Editor 11 June 2019; revised manuscript received 6 October 2019; published ahead of production 28 October 2019; published online 16 January 2020.

¹University of Science and Technology of China, School of Earth and Space Sciences, Hefei, China. E-mail: xinmwu@ustc.edu.cn (corresponding author).

²The University of Texas at Austin, Bureau of Economic Geology, Austin, Texas, USA. E-mail: zhichenggeng@utexas.edu; yzshi08@utexas.edu; nam-pham@utexas.edu; sergey.fomel@beg.utexas.edu.

³University of Lorraine, Nancy, France. E-mail: guillaume.caumon@ensg.univ-lorraine.fr.

© 2020 Society of Exploration Geophysicists. All rights reserved.

lighting the reflection discontinuities in a seismic image. However, measuring reflection discontinuity alone is insufficient to distinguish faults from other discontinuous features such as noise and stratigraphic features (Hale, 2013). Therefore, some postprocessing steps such as ant tracking (Pedersen et al., 2002, 2003), fault-oriented smoothing (Wu and Zhu, 2017), and optimal surface voting (Wu and Fomel, 2018a) are proposed to further enhance fault-related features, while suppressing nonfault noise in a fault attribute image. These postprocessing methods, however, highly depend on the input fault attribute and therefore may yield much different results with different input attributes.

To assist in seismic horizon interpretation, various methods such as phase-unwrapping (Stark, 2003; Wu and Zhong, 2012), waveform classification (Figueiredo et al., 2007, 2014, 2015), slope-based methods (Lomask et al., 2006; Fomel, 2010; Parks, 2010; Wu and Hale, 2013, 2015; Zinck et al., 2013; Monniron et al., 2016), and multigrid correlation (Wu and Fomel, 2018b) have been proposed to automate the horizon mapping from a seismic image. These methods, however, face a common challenge of dealing with complex structures such as intensive crossing faults and complicated folding, which requires a globally optimal correlation of all the structure elements in a seismic image.

Recently, some authors propose to improve automatic seismic fault (e.g., Huang et al., 2017; Di et al., 2018; Guo et al., 2018; Wu et al., 2018, 2019; Zhao and Mukhopadhyay, 2018) and horizon (Lowell and Paton, 2018; Wu and Zhang, 2018) interpretation by using convolutional neural networks (CNNs), which have shown excellent performance in common image processing tasks (e.g., Ren et al., 2015; Ronneberger et al., 2015; He et al., 2016, 2017; Badrinarayanan et al., 2017). However, the main limitation of applying CNNs in geoscience problems including the seismic structural interpretation is to prepare rich training data sets with reliable labels as discussed by Bergen et al. (2019).

In this paper, we propose a workflow to automatically create numerous realistic structure models, synthetic seismic images, and corresponding structural labels, which are further used to train CNNs for structural interpretation in field seismic images. We use vertical shearing fields and volumetric vector fields, respectively, to simulate realistic folding and faulting structures in the structure models and synthetic training data sets. We parameterize the shearing and volumetric vector fields with a set of variables, which are all randomly chosen within some reasonable ranges to create numerous models

and training data sets with various structures. By using the automatically generated synthetic seismic images and the ground truth labels of structures simulated in the images, we train a simplified U-net (Wu et al., 2019) and a deep CNN with an encoder-decoder architecture followed by a refinement network (Geng et al., 2019), respectively, for detecting faults and computing horizons or relative geologic time (RGT) volumes from seismic images. Multiple field examples show that the CNNs, trained with only synthetic data sets, effectively learn from the simulated structures to recognize real structures within field seismic images.

BUILDING STRUCTURAL MODELS

To build a folded and faulted structural model, we start with an initial model with all flat layers in the (X, Y, Z) space as shown in Figure 1a. We then sequentially add folding and faulting structures in this 3D model. Different from the conventional model building (e.g., Caumon et al., 2009; Georgsen et al., 2012) guided by specific geologic and geophysical data sets, we simulate generally realistic structures in a model by using randomly chosen parameters because we expect to automatically create numerous models, not a single model that fits some specific geologic knowledge.

Folding

We simulate the folding in a 3D structure model by vertically shearing the flat model, and the shearing is defined by a combination of two shift fields $S_1(X, Y, Z)$ and $S_2(X, Y, Z)$. The first shift field $S_1(X, Y, Z)$ is defined as a linear function as follows:

$$S_1(X, Y, Z) = aX + bY + c_0, \quad (1)$$

where a , b , and c_0 are randomly chosen within some predefined ranges. This linear shift field is used to create purely dipping structures in the model. The dipping angles in the x - and y -directions, respectively, are determined by the two parameters a and b , which are limited in the range of $[-0.25, 0.25]$ to avoid extremely dipping structures (after combining with S_2). We choose $c_0 = -aX_c - bY_c$ so that the center trace (X_c, Y_c, Z) of the model is not shifted.

Purely dipping structure is not sufficient to simulate realistic folding structures in a model. Therefore, we further add bended structures in the model by using the following vertical shift field:

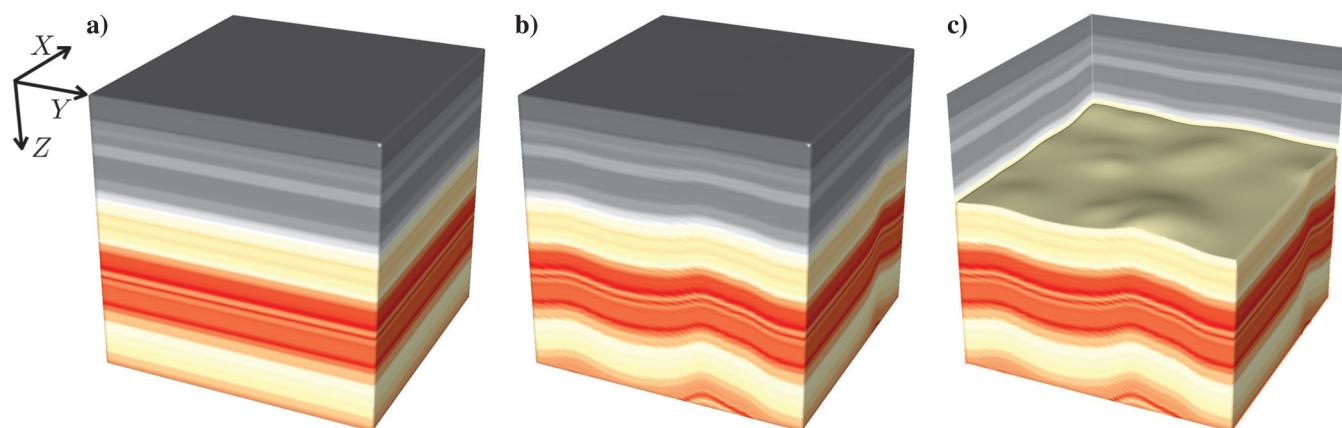


Figure 1. (a) A starting model with flat layers, (b) a folded model, and (c) a cut-away view of the folded model.

$$S_2(X, Y, Z) = \frac{1.5}{Z_{\max}} Z \sum_{k=1}^{k=N} b_k e^{\frac{(X-c_k)^2 + (Y-d_k)^2}{2\sigma_k^2}}, \quad (2)$$

which is defined by a combination of N 2D Gaussian functions and a linear scalar function $(1.5/Z_{\max})Z$. Each k th Gaussian function is defined by its center position (c_k, d_k) , half-width σ_k , and amplitude b_k , which are all randomly chosen. We recommend to choose a small b_k for a small half-width σ_k to avoid generating very sharp bending structures in the model. With the summation of the N Gaussians, we are able to create various shapes (not necessary a Gaussian shape) of bend structures. With the linear scalar function, we assume the bending extent of the structures decreases in the vertical direction (z -axis) from the bottom to the top.

By using the two types of vertical shearing shift fields $S_1(X, Y, Z)$ and $S_2(X, Y, Z)$, we translate the samples from the flat space $((X, Y, Z)$ in Figure 1a) to the folded space $((\tilde{X}, \tilde{Y}, \tilde{Z})$ in Figure 1b):

$$\begin{bmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{Z} \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z + S_1(X, Y, Z) + S_2(X, Y, Z) \end{bmatrix}. \quad (3)$$

Figure 1c shows a cutaway view of the folded model (Figure 1b), in which we observe various shapes of bend structures inside the model. By randomly choosing the parameters of the shift fields $S_1(X, Y, Z)$ and $S_2(X, Y, Z)$, we are able to automatically generate numerous folded models with all different bend structures.

Faulting

In addition to the folding structure, faulting is another important type of structure in a model. We simulate faulting structures in a structure model by using volumetric vector fields as discussed by Georgsen et al. (2012). Instead of modeling a specific faulting guided by seismic interpretation as discussed by Georgsen et al. (2012), we simulate various and generalized faulting structures by using randomly chosen faulting parameters.

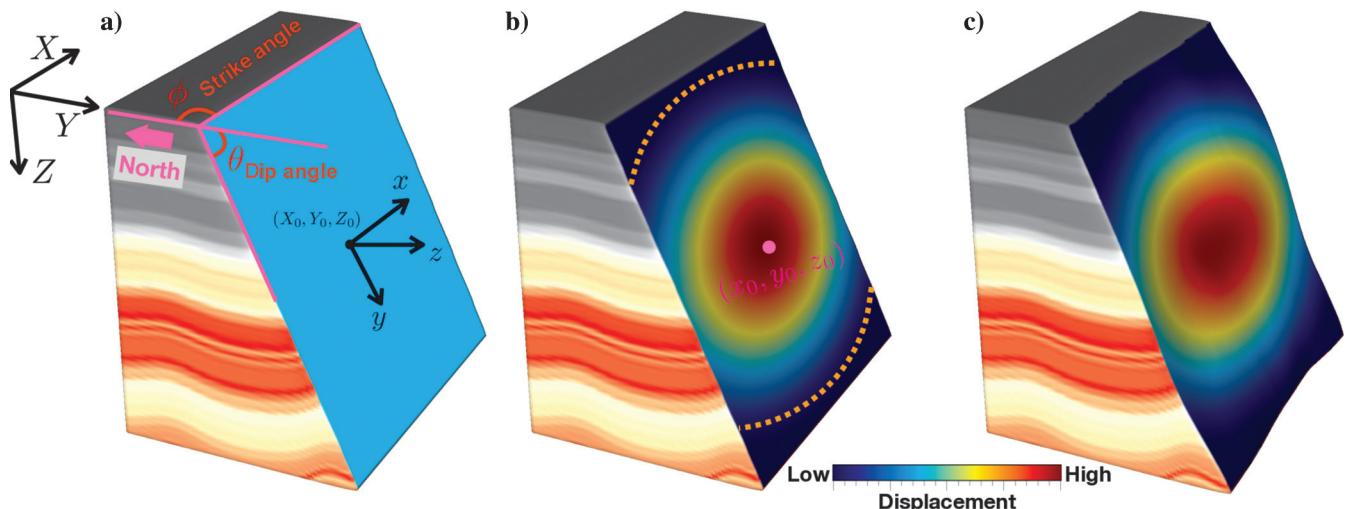


Figure 2. A combination of a strike angle ϕ , a dip angle θ , and a reference point (X_0, Y_0, Z_0) is chosen to define a (a) 3D fault plane, on which, (b) an elliptic displacement field is defined. (c) This fault plane is further bended to obtain a realistic fault surface.

Fault plane

To simulate faulting in the structure model, we start with defining a fault plane by randomly choosing a reference point (X_0, Y_0, Z_0) , a strike angle ϕ ($\phi \in [0^\circ, 360^\circ]$), and a dip angle θ ($\theta \in (0^\circ, 90^\circ)$) as shown in Figure 2a. We then define a local coordinate (x, y, z) with the origin (x_0, y_0, z_0) at the reference point, the x -axis in the strike direction, the y -axis in the dip direction, and the z -axis in the normal direction. With this definition, we can transform points from the global coordinates to the local coordinates as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}, \quad (4)$$

where we first translate the points relative to the reference point (X_0, Y_0, Z_0) , and then rotate the coordinates by using the rotation matrix

$$\mathbf{R} = \begin{bmatrix} \sin \phi & \cos \phi & 0 \\ \cos \phi \cos \theta & -\sin \phi \cos \theta & \sin \theta \\ \cos \phi \sin \theta & \sin \phi \sin \theta & -\cos \theta \end{bmatrix}. \quad (5)$$

Fault displacement

As suggested by some researchers (e.g., Walsh and Watterson, 1987; Georgsen et al., 2012), we define the displacement field $d(x, y; z = 0)$ on the fault plane as an ideal elliptic function with the center point at the origin (x_0, y_0, z_0) , the longest diameter l_x in the strike direction, and the shortest diameter l_y in the dip direction:

$$d(x, y, z = 0) = 2d_{\max}[1 - r(x, y)] \times \sqrt{\frac{[1 + r(x, y)]^2}{4} - r^2(x, y)}, \quad (6)$$

where l_x , l_y , and the maximum displacement d_{\max} at the center point are all randomly chosen. The term $r(x, y)$ is the normalized radial distance from the center point:

$$r(x, y) = \sqrt{\left(\frac{x - x_0}{l_x}\right)^2 + \left(\frac{y - y_0}{l_y}\right)^2}. \quad (7)$$

As displayed in color in Figure 2b, the displacement is maximum at the center point and decreases in all directions until dying out at the tip line (the dashed orange curve in Figure 2b) $r(x, y) = 1$. In practice, the displacement trend on a fault surface is not necessarily an ideal elliptic shape as defined in equation 6. We can further perturb the elliptic shape to an arbitrarily irregular shape as suggested by Barnett et al. (1987) and Walsh and Watterson (1989).

Perturb a fault surface

In practice, a 3D fault is rarely an ideal plane as shown in Figure 2a and 2b. We further perturb the fault plane to obtain a more realistic fault surface that is curved as shown in Figure 2c. To perturb such a curved surface, we first randomly generate M perturbation points $((x_k, y_k, z_k), k = 1, 2, \dots, M)$ near the fault plane as displayed as the small purple cubes in Figure 3b. We then interpolate a smooth and curved surface $z = f(x, y)$ (Figure 3c) from these seed points by using the biharmonic spline interpolation method (Sandwell, 1987).

In this way, we are able to simulate numerous curved fault surfaces as the one in Figure 3c. Figure 3d shows a listric fault surface (with relative low-dip angles) that is generated by using a different set of perturbation seed points (displayed as the small purple cubes).

Volumetric displacement field

To create faulting in the 3D structural model, we use a volumetric displacement field, which is a 3D vector field defined around the fault surface (Georgsen et al., 2012; Laurent et al., 2013). Such a volumetric displacement field $\mathbf{D}(x, y, z) = (D_x, D_y, D_z)$ contains three components that are defined in the local coordinates. Because the displacements in the strike direction (x -axis) are often hardly visible in a structure model, we therefore assume the strike-displacement $D_x(x, y, z) = 0$ and define nonzero volumetric displacements only in the dip (y -axis) and normal (z -axis) directions.

We extrapolate the displacements $d(x, y; z = 0)$ (equation 6) from the fault surface to estimate the dip-displacement component

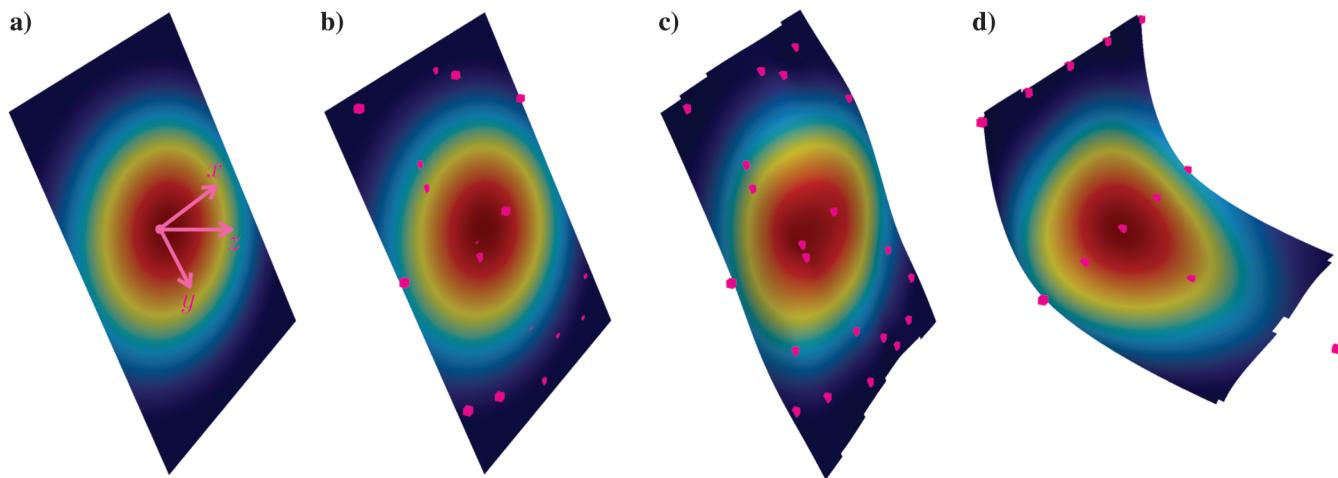


Figure 3. A workflow of (a and b) perturbing a 3D fault plane to obtain (c) a curved or (d) a listric fault surface.

$D_y(x, y, z)$ in the hanging-wall and foot-wall blocks around the fault surface. As suggested by Barnett et al. (1987), the displacement decreases away from the fault surface ($z = f(x, y)$) in the z -axis direction and it reaches zero at the reverse drag radius γ . For any point (x, y, z) in the hanging-wall block ($f(x, y) \leq z \leq \gamma + f(x, y)$), we compute the dip-displacement $D_y(x, y, z)$ as

$$D_y(x, y, z) = \lambda \cdot d(x, y; z = 0) \cdot \alpha(x, y, z), \quad (8)$$

where $\alpha(x, y, z)$ is a nonlinear scalar function that decreases away from the fault surface $z = f(x, y)$ along the z -axis direction as suggested by Cardozo et al. (2008):

$$\alpha(x, y, z) = \left(1 - \frac{|z - f(x, y)|}{\gamma}\right)^2. \quad (9)$$

For any point (x, y, z) in the foot-wall block ($f(x, y) - \gamma \leq z \leq f(x, y)$), the dip-displacement $D_y(x, y, z)$ is computed as

$$D_y(x, y, z) = (\lambda - 1) \cdot d(x, y; z = 0) \cdot \alpha(x, y, z). \quad (10)$$

The λ in equations 8 and 10 is a ratio ($0 < \lambda < 1$) of the hanging-wall and foot-wall displacements.

The color in Figure 4b represents the dip-displacement field $D_y(x, y, z)$ computed by using equations 8 and 10. Positive displacements in the hanging-wall block and negative displacements in the foot-wall block, as shown in Figure 4b, relatively shift the fault blocks to produce a normal fault in the structure model. A reverse fault can be produced by reversing the signs of the displacements in the hanging-wall and foot-wall blocks.

Because the fault is a curved surface, the displacements in the z -direction must be nonzero as well to make sure that the fault blocks move seamlessly along the fault surface. We therefore compute $D_z(x, y, z)$ in the hanging-wall and foot-wall blocks as follows (Georgsen et al., 2012):

$$D_z(x, y, z) = f(x, y + D_y(x, y, z)) - f(x, y), \quad (11)$$

which is the variation of the fault surface with respect to the dip displacement $D_y(x, y, z)$. The color in Figure 4c shows the z -direction displacements $D_z(x, y, z)$ calculated using equation 11. To better

visualize the volumetric displacements in Figure 4b and 4c, we display the cutaway views of the same volumes in Figure 5. In Figure 5c, the displacements in the z -direction consistently follow the fault surface trend because we observe higher absolute displacements near areas where the fault surface has higher curvatures.

With the volumetric displacement fields $D_y(x, y, z)$ and $D_z(x, y, z)$ defined in the local coordinates, we generate faulting in the folded structure model $(\tilde{X}, \tilde{Y}, \tilde{Z})$ by the following three steps, each being highlighted in red:

(I) Transform the coordinates from $(\tilde{X}, \tilde{Y}, \tilde{Z})$ to (x, y, z)

$$\mathbf{R} \left(\begin{bmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{Z} \end{bmatrix} - \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \right), \quad (12)$$

(II) Apply fault displacements in the local coordinates (x, y, z)

$$\mathbf{R} \left(\begin{bmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{Z} \end{bmatrix} - \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ D_y(x, y, z) \\ D_z(x, y, z) \end{bmatrix}, \quad (13)$$

(III) Transform back from (x, y, z) to $(\tilde{X}, \tilde{Y}, \tilde{Z})$

$$\mathbf{R}^\top \left(\mathbf{R} \left(\begin{bmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{Z} \end{bmatrix} - \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ D_y(x, y, z) \\ D_z(x, y, z) \end{bmatrix} \right) + \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix}. \quad (14)$$

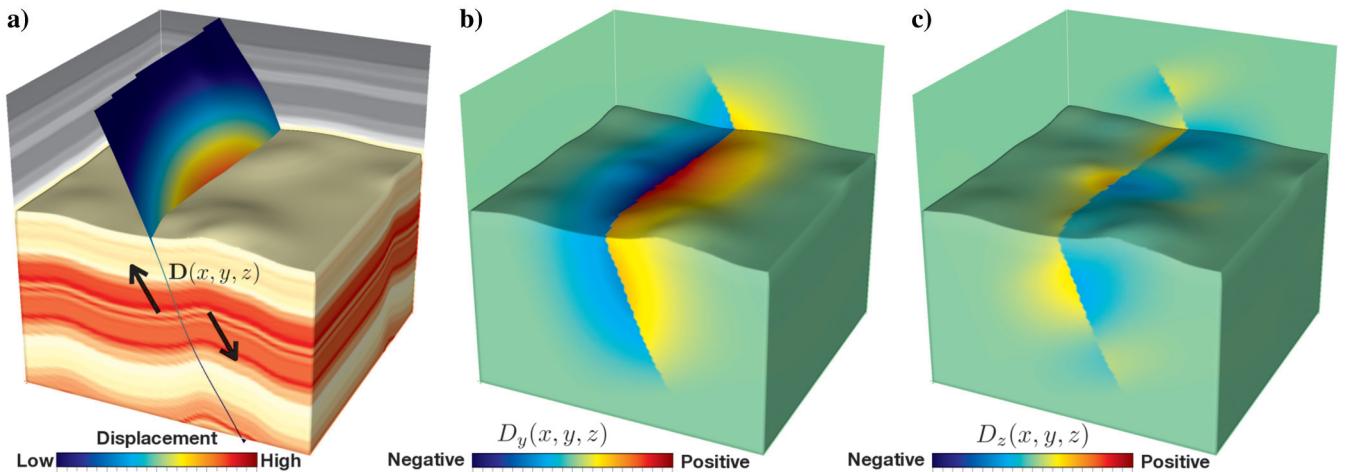


Figure 4. From the displacement field (a) defined on the fault surface, we extrapolate a 3D vector field of dip slips (in the local coordinates (x, y, z)) to create dip-slip faulting in the model. This vector field contains only two components in the (b) y - and (c) z -directions and we assume the strike-slip (x -direction) is zero.

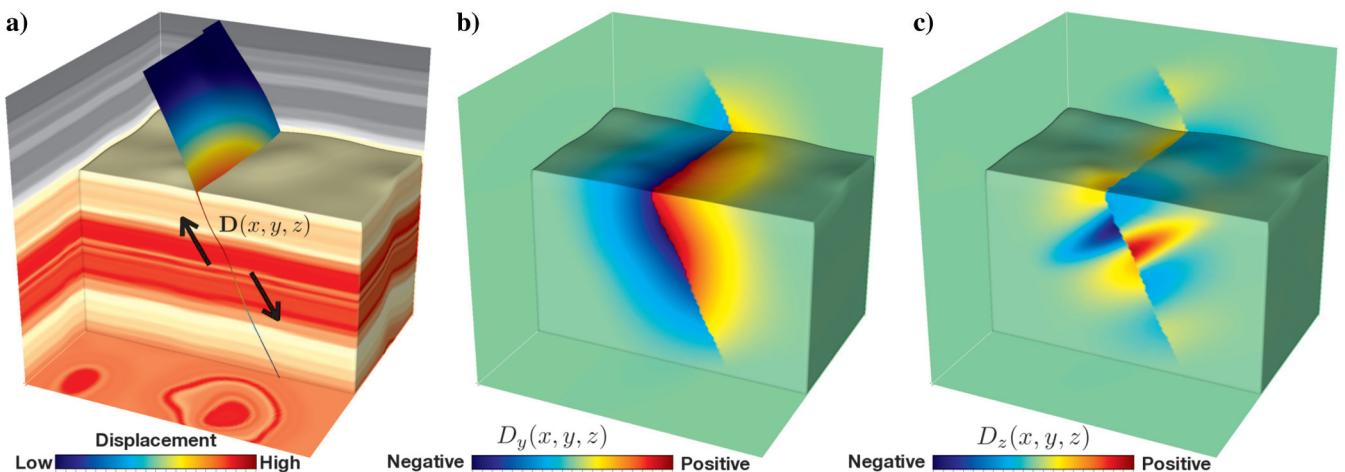


Figure 5. A cut-away view of the same volumes in Figure 4.

By using this three-step processing, we apply the volumetric displacements to the folded model (Figures 4a and 5a) and obtain a folded and faulted model shown in Figure 6a and 6b, in which the hanging-wall and foot-wall blocks are relatively shifted along the fault surface in the fault-dip direction. In addition to the relative shifting, we can also observe the reverse drag structure (the concave layers in Figure 6b compared to flat layers before faulting in Figure 4a) due to the faulting. Similarly, we can also simulate the normal drag of normal faulting and the reverse and normal drag of reverse faulting as discussed by Grasemann et al. (2005).

By randomly choosing the fault parameters discussed in this faulting workflow, we are able to generate various faulting patterns in the structure model. For example, we generated a listric fault with a totally different faulting pattern in the structure model as shown in Figure 6, in which we fixed the foot-wall block and we only move the hanging-wall block. In addition, we can apply a sequence of faulting to generate multiple faults within the structure model.

We have discussed a general way to parameterize folding and faulting in a 3D structure model. All of the folding and faulting parameters are summarized in Figure 7. A specific set of these parameters defines a unique folded and faulted model as shown in Figure 6b or 6c. By randomly choosing these parameters (each parameter has multiple options), we are able to create numerous unique structure models. This is important for our next step of creating rich data sets and labels to train CNNs for seismic structural interpretation.

SYNTHETIC SEISMIC IMAGES AND STRUCTURAL LABELS

The CNN has been proven to be the most powerful method in image processing tasks such as image classification (Krizhevsky et al., 2012; Zeiler and Fergus, 2014; He et al., 2016), segmentation (Ronneberger et al., 2015; Badrinarayanan et al., 2017), and object detection (Girshick et al., 2014; Ren et al., 2015; He et al., 2017). Recently, the CNN has also been introduced to seismic image processing and shows promising performance in automatic interpretation of faults (e.g., Wu et al., 2018, 2019; Zhao and Mukhopadhyay, 2018), salt bodies (Shi et al., 2019), channels (Pham et al., 2019), and horizons (Geng et al., 2019).

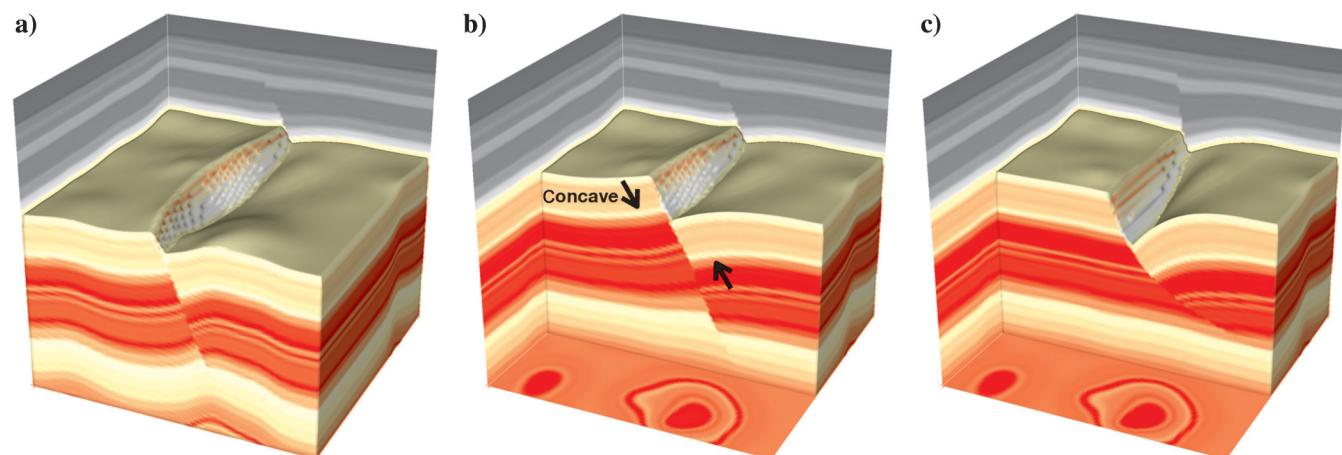


Figure 6. The folded model is further faulted in (a and b) by using the vector field of dip slips in Figure 4b and 4c. Similarly, we can create the listric faulting in (c).

The main limitation of applying CNN in seismic image processing remains in preparing rich training data sets and the corresponding labels. Manually labeling 3D seismic images can be highly labor intensive and subjective. It is hardly possible to manually label all of the geologic features (e.g., faults, horizons, and channels) within a 3D seismic image, and different interpreters may provide quite different interpretation results. Incomplete or inaccurate labeling in the training data sets will mislead the training of the CNN, and then the trained CNN cannot make reliable predictions. To solve this problem, we use the proposed workflow of folding and faulting to automatically create numerous synthetic seismic images and the known structural labels to train CNNs for structural interpretation in field seismic images.

To create a synthetic seismic image, we begin with an initially flat reflectivity model (Figure 8a), which is generated by horizontally extending a 1D trace of random reflectivity values. We then apply the proposed workflow of folding and faulting to obtain a folded and faulted reflectivity model shown in Figure 8b. We further convolve the reflectivity model with a wavelet (with a randomly chosen peak frequency) and obtain a synthetic seismic image shown in Figure 8c. As discussed by Wu et al. (2019), applying the convolution after the folding and faulting is helpful to simulate a more realistic fault in the seismic image because the convolution smears (Figure 8c) the sharp fault in Figure 8b. Finally, we add some extent of random noise to make the synthetic seismic image even more realistic as shown in Figure 8d.

Folding parameters		Faulting parameters	
a, b	Parameters of linear shearing	$\phi, \theta, (X_0, Y_0, Z_0)$	Parameters of a fault plane
b_k	Amplitudes of N Gaussians	(x_k, y_k, z_k)	Coordinates of M seed points for fault surface perturbation
(c_k, d_x)	Center coordinates of N Gaussians	d_{\max}, l_x, l_y	Parameters of displacements on the fault plane
σ_k	Half-width of N Gaussians	γ, λ	Parameters of volumetric displacements
Total : 2+4N parameters		Total : 5+3M+3+2 parameters	

Figure 7. A summary of the folding and faulting parameters.

Because the folding and faulting created in the synthetic seismic image (Figure 8d) are well-defined, we can then automatically and accurately label all of the structures within the seismic image. The most convenient way to record the structures during the folding and

faulting process is to define a monotonic volumetric function (Cherpeau et al., 2010) or an RGT volume (Stark, 2003, 2004; Wu and Zhong, 2012) as shown in Figure 9a. At the same time of the folding and faulting of the reflectivity model, we apply the same processing

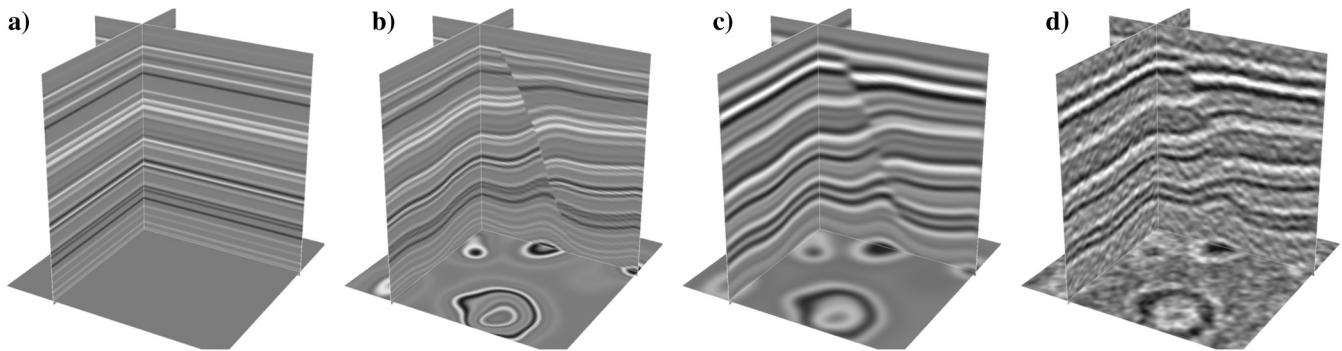


Figure 8. By applying the same workflow of folding and faulting to (a) an initial 3D flat reflectivity model, we are able to create (b) a folded and faulted model, which is further convolved with a Ricker wavelet to obtain (c) a synthetic seismic image. (d) We further add some random noise to make a more realistic seismic image.

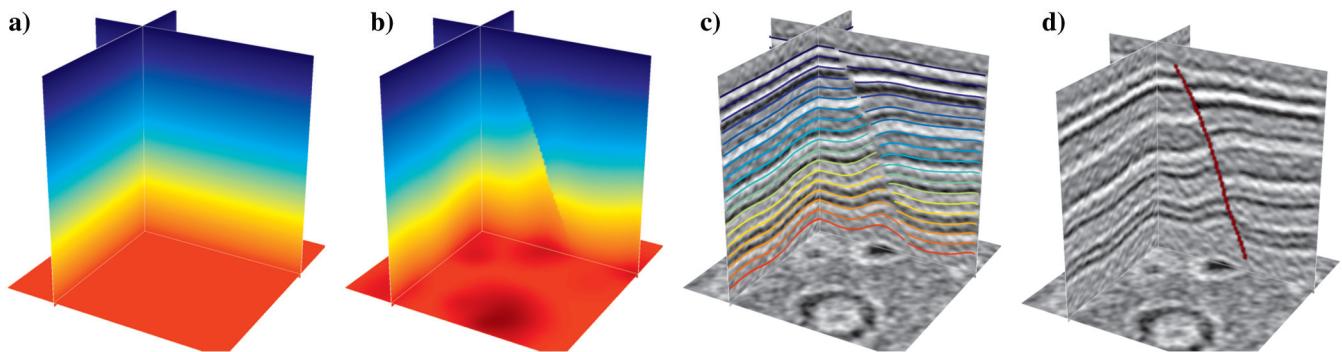


Figure 9. At the same time of folding and faulting the reflectivity model (Figure 8b), we apply the same processing to (a) an initial monotonic volumetric function or RGT volume to obtain (b) a folded and faulted RGT volume that implicitly contains all the structural information in the corresponding seismic image (Figure 8c). From the RGT volume, the structural labels of (c) horizons and (d) faults, respectively, can be extracted as contours and lateral discontinuities of the RGT volume.

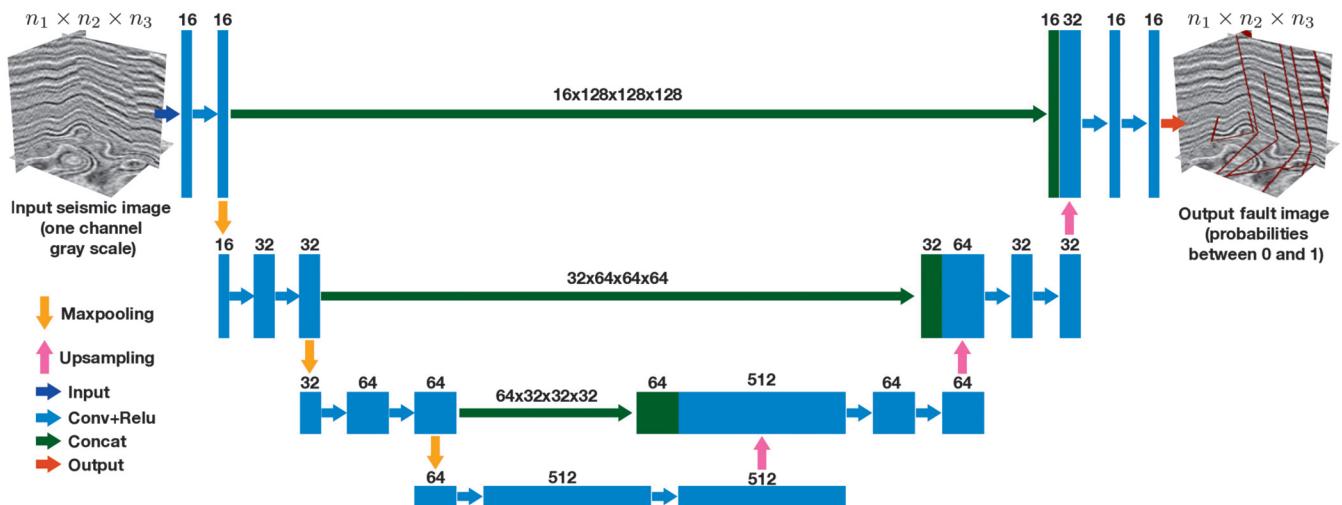


Figure 10. A CNN for fault segmentation (Wu et al., 2019).

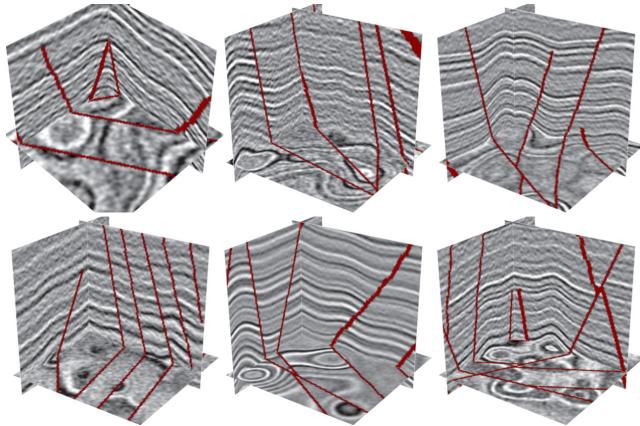


Figure 11. Automatically generated seismic images and the corresponding fault labels.

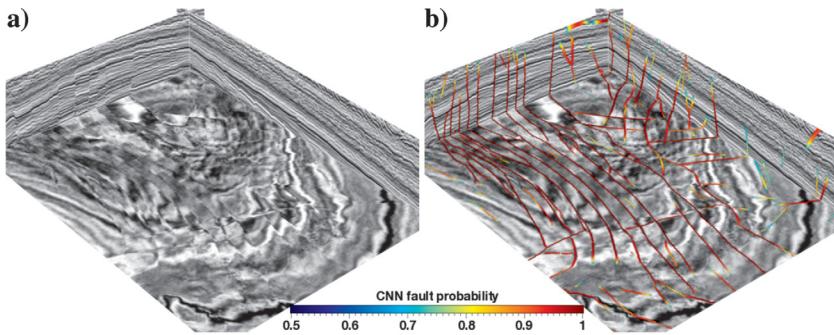


Figure 12. Fault detection in a subset of the F3 seismic data using the CNN trained by only synthetic data sets.

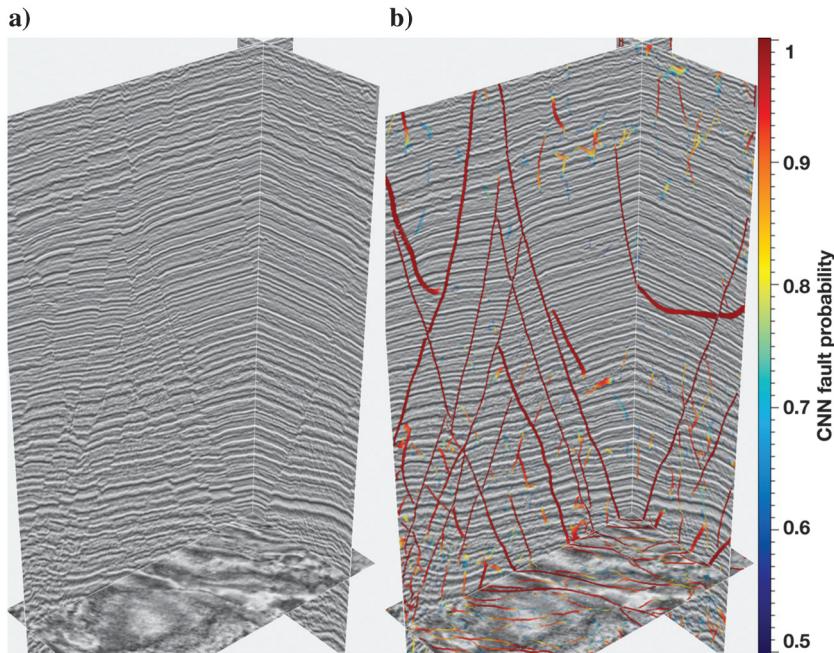


Figure 13. Fault detection in a subset of the Westcam seismic data using the CNN trained by only synthetic data sets.

to obtain a consistently folded and faulted RGT volume shown in Figure 9b. In this way, all the folding and faulting structures within the seismic image (Figure 8d) are implicitly recorded in the corresponding RGT volume (Figure 9b). All the seismic horizons are implicitly represented as isovalue surfaces (Figure 9c) of the RGT volume. The fault positions (Figure 9d) are implicitly indicated as lateral discontinuities along the isovalue surfaces. The fault displacements are indicated by the dislocations of the isovalue surfaces. This implicit way is especially convenient to record complicated structures when the model is dislocated by multiple intersecting faults.

CNN FOR SEISMIC STRUCTURAL INTERPRETATION

To evaluate the validness and representativeness of the simulated structures, we apply the automatically generated synthetic seismic images and the corresponding structure labels to train deep CNNs for structural interpretation in field seismic images. We first train a CNN for fault detection in 3D seismic images, and then we train another CNN to estimate RGT from 2D seismic images.

Train a CNN for fault segmentation

We consider fault detection in a seismic image as an image segmentation problem of labeling ones at fault locations while there are zeros elsewhere. We perform the fault segmentation by using a simplified U-Net (Wu et al., 2019) as shown in Figure 10. In this U-shape architecture, the left contracting path contains three steps. Each step sequentially contains two $3 \times 3 \times 3$ convolutional layers, a ReLU activation, and a $2 \times 2 \times 2$ max pooling with stride 2. Each step in the right expansive path sequentially contains a $2 \times 2 \times 2$ upsampling operation with stride 2, a concatenation with features from the left path, two $3 \times 3 \times 3$ convolutional layers, and a ReLU activation.

To train this fault segmentation network, we used the proposed workflow to automatically create 200 $128 \times 128 \times 128$ synthetic seismic images and the corresponding fault labeling images with ones at the fault positions while there are zeros elsewhere. Figure 11 shows six of the training seismic images displayed with the corresponding fault labeling images, in which we have created multiple faults in each of the seismic images. In this paper, we used training images with fixed dimensions of $128 \times 128 \times 128$; however, the size of the training and test images can vary. The only limitation is that each dimension needs to be divisible by eight because we have three times of downsampling (we reduce the dimension of an input by half) and upsampling (we double the dimension of an input) included in the U-net architecture. We actually recommend to use larger training seismic images

if the graphics processing unit (GPU) memory allows. Further details of the CNN and training processing are discussed by Wu et al. (2019).

Figures 12, 13, 14, and 15 show the fault predictions in four field seismic images (acquired at totally different surveys) by using

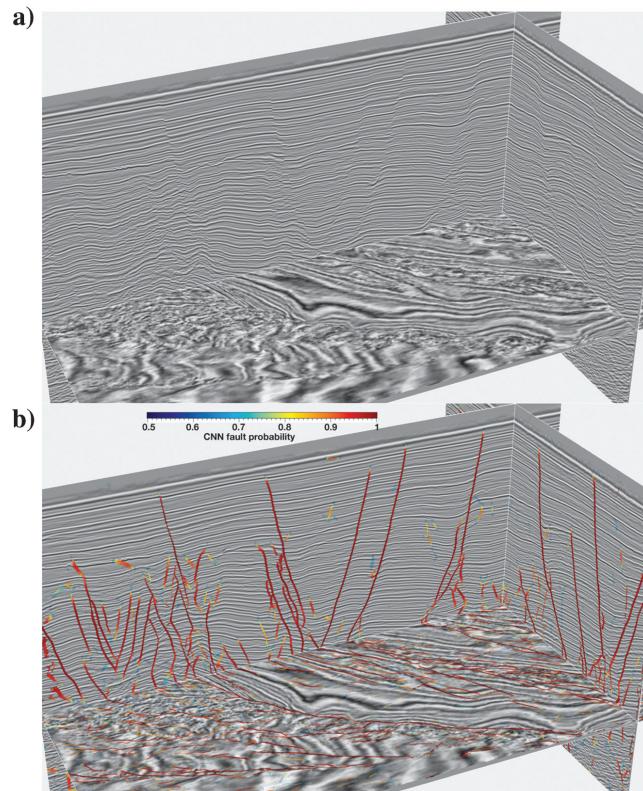


Figure 14. Fault detection in a subset of the Opunake-3D seismic data using the CNN trained by only synthetic data sets.

the trained CNN model. Although trained with only synthetic data sets, the CNN model works pretty well to compute clean and accurate fault detections from field seismic images that are new to the CNN model. This indicates that the structures, simulated by using the proposed workflow, are realistic and representative enough for the CNN to effectively learn fault detection in general seismic images. Note that in the examples shown in Figures 13–15, we can observe significant variations of reflection patterns (that may correspond to frequency variations) from shallow to deep in the seismic images. The CNN, trained by synthetic data sets, still successfully detects the faults apparent in the whole vertical seismic sections.

Train a CNN for RGT and seismic horizons

RGT estimation from a seismic image is a more challenging task than is fault detection. Several types of methods including phase unwrapping (Stark, 2003; Wu and Zhong, 2012) and slope-based methods (Lomask et al., 2006; Fomel, 2010; Parks, 2010; Wu and Hale, 2013, 2015; Zinck et al., 2013) have been proposed to perform automatic RGT estimation. However, estimating geologically consistent RGT values across faults, especially the complicated crossing faults, remains a highly challenging problem for all of the methods. RGT estimation from a seismic image is a more challenging regression problem than fault segmentation for the CNN as well; therefore, we use a more complicated CNN (Geng et al., 2019) with deeper layers (Figure 16a) than the fault segmentation CNN (Figure 10). As shown in Figure 16a, the CNN consists of an encoder-decoder architecture followed by a refinement module. The encoder is a 50-layer residual network (He et al., 2016), which contains five sections, each consisting of multiple residual blocks and downsample the inputs by one-half scale. As shown in Figure 16b, each residual block contains three convolutional layers implemented by 1×1 , 3×3 , and 1×1 filters, respectively. The decoder contains five up-projection blocks (Laina et al., 2016) that upsample the feature maps back to the same size as the input seismic image. Each up-projection block is an upsampling

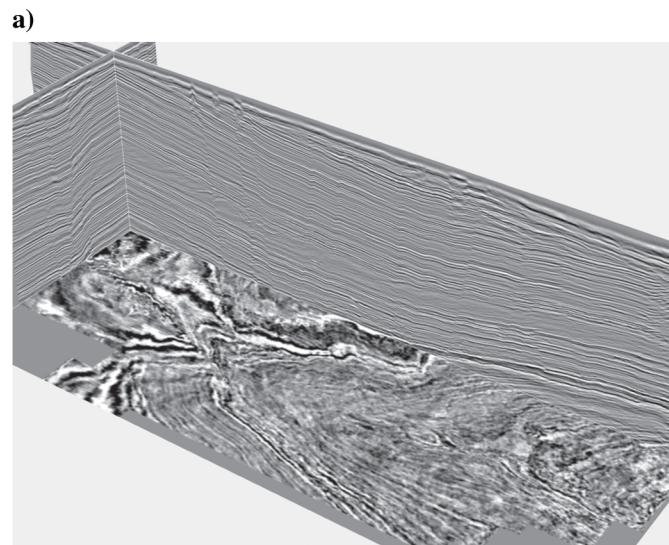


Figure 15. Fault detection in a subset of the Kerry-3D seismic data using the CNN trained by only synthetic data sets.

residual block as shown in Figure 16c. The followed refinement module consists of three convolutional layers, which makes a final prediction of an RGT volume from the encoder-decoder features. More details of the deep CNN (Figure 16) are discussed by Geng et al. (2019).

To train this CNN for RGT estimation, we created 100 3D synthetic seismic images and the corresponding RGT volumes by using the proposed workflow of folding and faulting. Figure 17 shows six of these seismic images displayed in gray and the corresponding RGT volumes displayed in color. Because the CNN is deep, we test it on only 2D data sets to save GPU memory and computational costs. Therefore, we extracted 2D sections from the 3D training data sets and apply some data augmentations (horizontal and vertical flipping) to the 2D sections to finally obtain approximately 2000 sets of training data sets.

Figure 18 shows the RGT predictions (Figure 18b) in three different field seismic images (Figure 18a) by using the CNN model trained with the automatically generated synthetic data sets. The CNN model works well to estimate accurate RGT volumes, even across the complex crossing faults, which is highly challenging for all conventional methods. The contours (Figure 18c) of the RGT volumes consistently follow seismic horizons that are complicatedly folded and faulted. The horizon dislocations across the faults provide a good estimation of fault displacements. Note that all of the field seismic images in Figure 18 share the same size (256×256) as the training seismic images. In applying the trained CNN model to a larger seismic image, some postprocessing of merging RGT subvolumes or recursively extracting horizons from multiple overlapping RGT subvolumes is required as discussed by Geng et al. (2019).

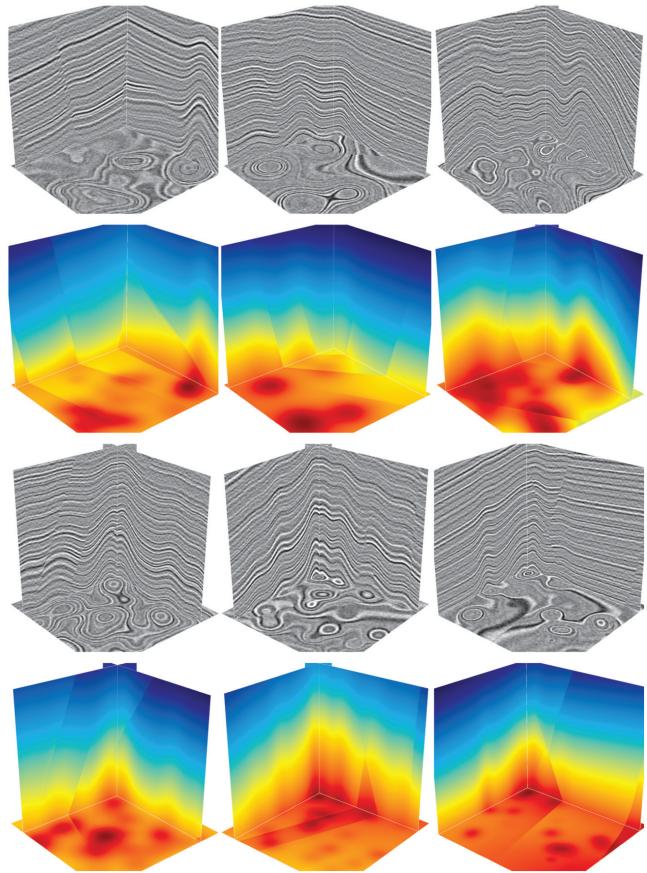


Figure 17. Automatically generated seismic images (displayed in gray) and the corresponding RGT volumes (displayed in color).

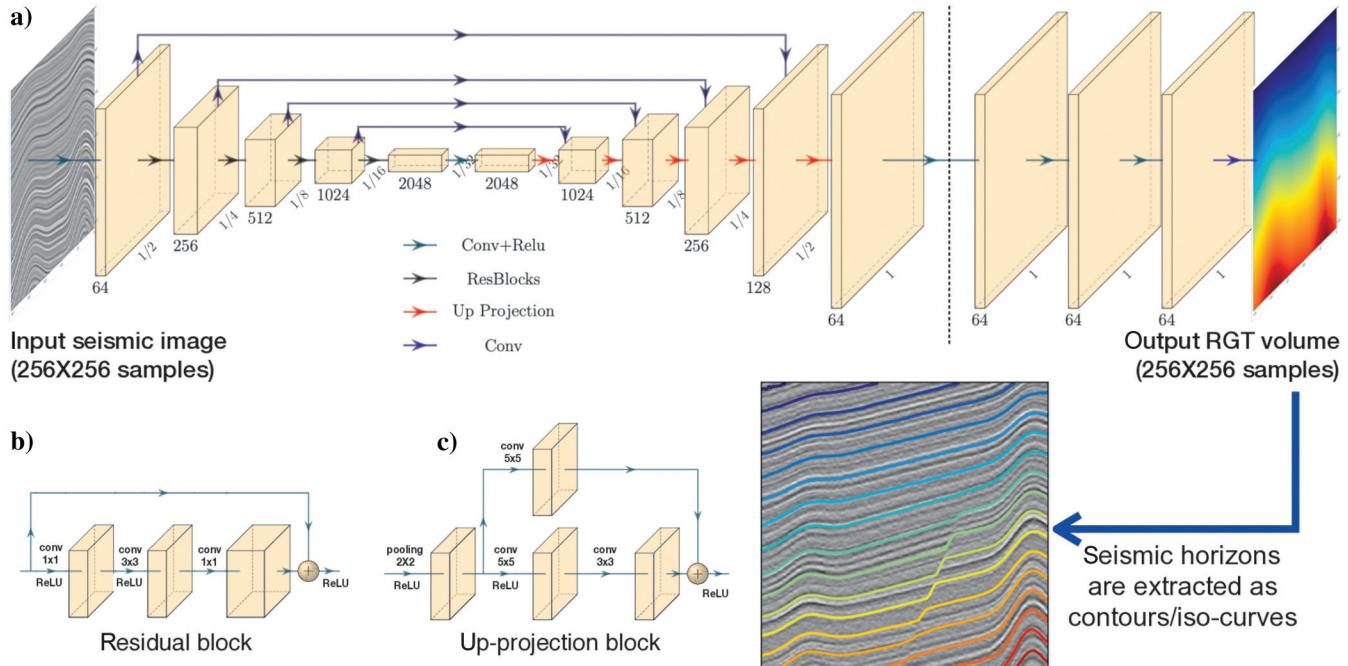


Figure 16. A deep CNN (Geng et al., 2019) to estimate an RGT volume from an input seismic image. From the estimated RGT volume, seismic horizons can be extracted as RGT contours as denoted by the colorful curves overlaid with the seismic image in the bottom-right figure.

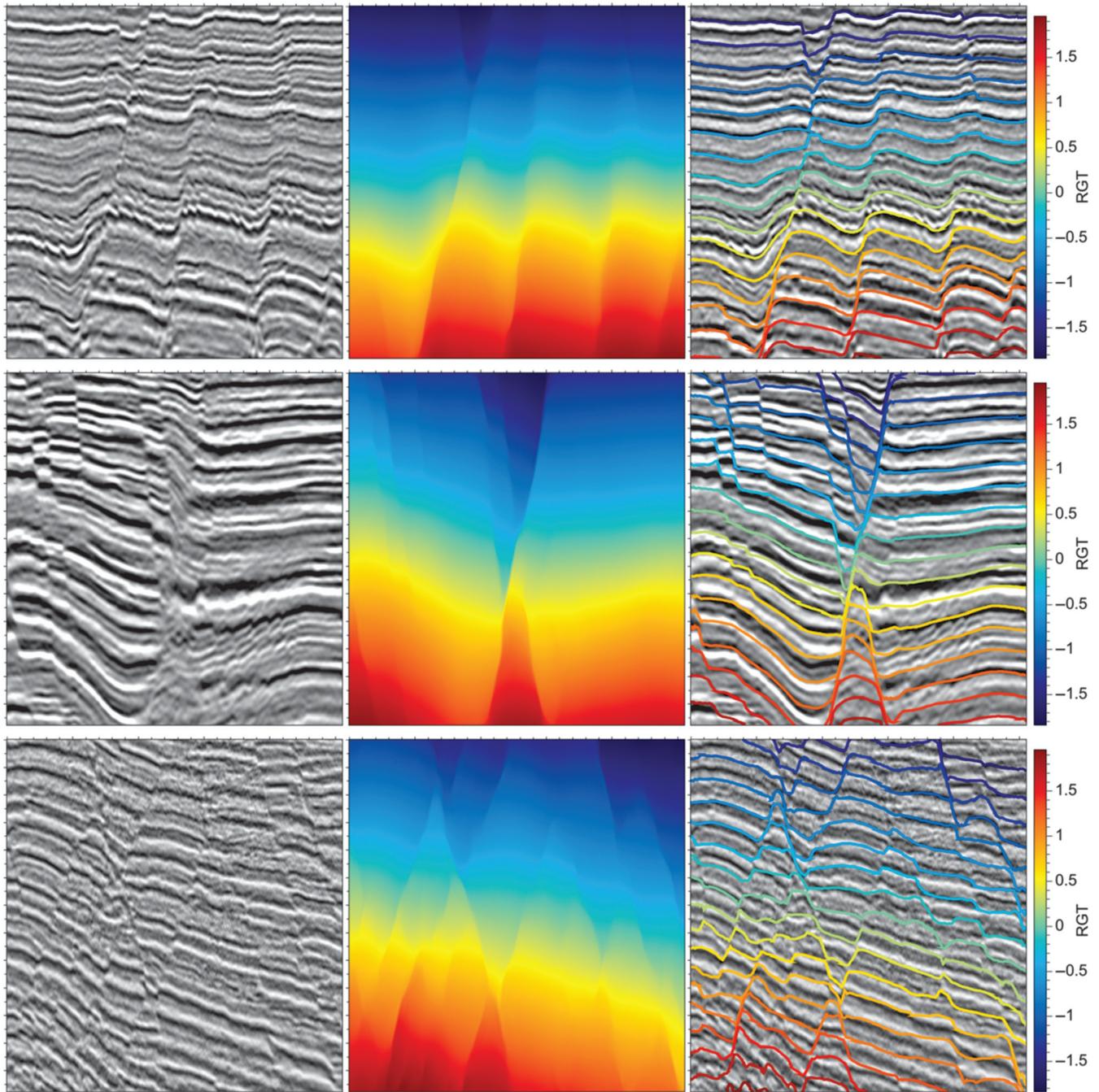


Figure 18. By applying the deep CNN (Figure 17), trained with only synthetic data sets, to (a) three different field seismic images, we are able to accurately estimate (b) RGT volumes from which (c) seismic horizons can be extracted as contours.

CONCLUSION

We have discussed a workflow to automatically simulate realistic and representative folding and faulting structures in a structure model. The folding and faulting are reasonably parameterized by multiple variables, which are randomly chosen to create numerous unique structure models. The main motivation of this work is to create rich synthetic seismic images and obtain the corresponding exactly accurate structural labels (without any human interpretation) for training deep CNNs. We have used the created synthetic

seismic images and structural labels to train CNNs for 3D fault segmentation and 2D RGT estimation. The trained CNNs worked well to provide excellent estimations of fault positions and RGT volumes from various field seismic images that were not seen by the CNNs during the training. This indicates that the simulated folding and faulting structures are realistic and representative for training CNNs to perform tasks of seismic structural interpretation.

In this paper, we have not yet included geobodies (e.g., channels, salt bodies, and igneous intrusions) and unconformities in simulating our structural models and synthetic seismic images. However,

the seismic structural interpretation, especially the RGT estimation, may be significantly complicated by the geobodies and unconformities. Therefore, we need to create even more realistic structure models with various geobodies and simulate stratigraphic models to train more generalized CNNs for seismic structural interpretation. In simulating the synthetic seismic images, we convolve a folded and faulted reflectivity model with a Ricker wavelet, which can ensure consistency between the synthetic seismic images and the known structural labels but fails to simulate the seismic migration errors or artifacts in the field seismic images. We have added random noise to the synthetic seismic images but have not figured out an effective way to add coherent noises as in the field images, which are often more challenging to deal with than the random noise in seismic interpretation. In addition, we may also need to take into account the lateral variations of rock properties and spatial variation of the wavelet phases and frequencies, which may also affect the structure features apparent in our simulated synthetic seismic images.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation of China under grant no. 41974121.

DATA AND MATERIALS AVAILABILITY

Data associated with this research are available and can be obtained by contacting the corresponding author.

REFERENCES

- Badrinarayanan, V., A. Kendall, and R. Cipolla, 2017, Segnet: A deep convolutional encoder-decoder architecture for image segmentation: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 2481–2495, doi: [10.1109/TPAMI.2016.253134](https://doi.org/10.1109/TPAMI.2016.253134).
- Barnett, J. A., J. Mortimer, J. H. Rippon, J. J. Walsh, and J. Watterson, 1987, Displacement geometry in the volume containing a single normal fault: *AAPG Bulletin*, **71**, 925–937.
- Baudouin, C., and J. Cartwright, 2008, The kinematics of reactivation of normal faults using high resolution throw mapping: *Journal of Structural Geology*, **30**, 1072–1084, doi: [10.1016/j.jsg.2008.04.008](https://doi.org/10.1016/j.jsg.2008.04.008).
- Bergen, K. J., P. A. Johnson, M. V. de Hoop, and G. C. Beroza, 2019, Machine learning for data-driven discovery in solid earth geoscience: *Science*, **363**, eaau0323, doi: [10.1126/science.aau0323](https://doi.org/10.1126/science.aau0323).
- Cardozo, N., P. Røe, H. Soleng, N. Fredman, J. Tveranger, and S. Schueller, 2008, A methodology for efficiently populating faulted corner point grids with strain: *Petroleum Geoscience*, **14**, 205–216, doi: [10.1144/1354-079308-738](https://doi.org/10.1144/1354-079308-738).
- Caumon, G., P. Collon-Drouaillet, C. Le Carlier de Veslud, S. Viseur, and J. Sausse, 2009, Surface-based 3D modeling of geological structures: *Mathematical Geosciences*, **41**, 927–945, doi: [10.1007/s11004-009-9244-2](https://doi.org/10.1007/s11004-009-9244-2).
- Cherpeau, N., G. Caumon, and B. Lévy, 2010, Stochastic simulations of fault networks in 3D structural modeling: *Comptes Rendus Geoscience*, **342**, 687–694, doi: [10.1016/j.crit.2010.04.008](https://doi.org/10.1016/j.crit.2010.04.008).
- Di, H., M. Shafiq, and G. AlRegib, 2018, Patch-level MLP classification for improved fault detection: 88th Annual International Meeting, SEG, Expanded Abstracts, 2211–2215, doi: [10.1190/segam2018-2996921.1](https://doi.org/10.1190/segam2018-2996921.1).
- Ellevset, S. O., R. Knipe, T. S. Olsen, Q. Fisher, and G. Jones, 1998, Fault controlled communication in the Sleipner Vest Field, Norwegian continental shelf: detailed, quantitative input for reservoir simulation and well planning: *Geological Society, London, Special Publications* 147, 283–297.
- Figueiredo, A., F. B. Silva, P. Silva, L. de O. Martins, R. L. Milidiú, and M. Gattass, 2015, A clustering-based approach to map 3D seismic horizons: 14th International Congress of the Brazilian Geophysical Society & EXPOGEF, 1166–1170.
- Figueiredo, A. M., M. Gattass, and F. Szenberg, 2007, Seismic horizon mapping across faults with growing neural gas: 10th International Congress of the Brazilian Geophysical Society & EXPOGEF, 1476–1481.
- Figueiredo, A. M., P. M. Silva, M. Gattass, F. B. Silva, and R. L. Milidiú, 2014, A seismic faces analysis approach to map 3D seismic horizons: 84th Annual International Meeting, SEG, Expanded Abstracts, 1501–1505, doi: [10.1190/segam2014-1382.1](https://doi.org/10.1190/segam2014-1382.1).
- Fomel, S., 2010, Predictive painting of 3D seismic volumes: *Geophysics*, **75**, no. 4, A25–A30, doi: [10.1190/1.3453847](https://doi.org/10.1190/1.3453847).
- Geng, Z., X. Wu, Y. Shi, and S. Fomel, 2019, Relative geologic time estimation using a deep convolutional neural network: 88th Annual International Meeting, SEG, Expanded Abstracts, 2238–2242, doi: [10.1190/segam2019-3214459.1](https://doi.org/10.1190/segam2019-3214459.1).
- Georgsen, F., P. Røe, A. R. Syversveen, and O. Lia, 2012, Fault displacement modelling using 3D vector fields: *Computational Geosciences*, **16**, 247–259, doi: [10.1007/s10596-011-9257-z](https://doi.org/10.1007/s10596-011-9257-z).
- Girshick, R., J. Donahue, T. Darrell, and J. Malik, 2014, Rich feature hierarchies for accurate object detection and semantic segmentation: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 580–587.
- Grasemann, B., S. Martel, and P. Cees, 2005, Reverse and normal drag along a fault: *Journal of Structural Geology*, **27**, 999–1010, doi: [10.1016/j.jsg.2005.04.006](https://doi.org/10.1016/j.jsg.2005.04.006).
- Guo, B., L. Li, and Y. Luo, 2018, A new method for automatic seismic fault detection using convolutional neural network: 88th Annual International Meeting, SEG, Expanded Abstracts, 1951–1955, doi: [10.1190/segam2018-2995894.1](https://doi.org/10.1190/segam2018-2995894.1).
- Hale, D., 2009, Structure-oriented smoothing and semblance: CWP Report 635.
- Hale, D., 2013, Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3D seismic images: *Geophysics*, **78**, no. 2, O33–O43, doi: [10.1190/geo2012-0331.1](https://doi.org/10.1190/geo2012-0331.1).
- He, K., G. Gkioxari, P. Dollár, and R. Girshick, 2017, Mask R-CNN: IEEE International Conference on Computer Vision (ICCV), 2980–2988.
- He, K., X. Zhang, S. Ren, and J. Sun, 2016, Deep residual learning for image recognition: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.
- Huang, L., X. Dong, and T. E. Clee, 2017, A scalable deep learning platform for identifying geologic features from seismic attributes: *The Leading Edge*, **36**, 249–256, doi: [10.1190/tle36030249.1](https://doi.org/10.1190/tle36030249.1).
- Knipe, R. J., G. Jones, and Q. Fisher, 1998, Faulting, fault sealing and fluid flow in hydrocarbon reservoirs: An introduction: Geological Society, London, Special Publications 147, vii–xxi.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks: Advances in Neural Information Processing Systems, 1097–1105.
- Kuszmir, N., and G. Karner, 2007, Continental lithospheric thinning and breakup in response to upwelling divergent mantle flow: Application to the Woodlark, Newfoundland and Iberia margins: Geological Society, London, Special Publications 282, 389–419.
- Laina, I., C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, 2016, Deeper depth prediction with fully convolutional residual networks: Proceedings of the 4th IEEE International Conference on 3D Vision (3DV), 239–248.
- Laurent, G., G. Caumon, A. Bouziat, and M. Jessell, 2013, A parametric method to model 3D displacements around faults with volumetric vector fields: *Tectonophysics*, **590**, 83–93, doi: [10.1016/j.tecto.2013.01.015](https://doi.org/10.1016/j.tecto.2013.01.015).
- Lomask, J., A. Guittot, S. Fomel, J. Claerbout, and A. A. Valenciano, 2006, Flattening without picking: *Geophysics*, **71**, no. 4, P13–P20, doi: [10.1190/1.2210848](https://doi.org/10.1190/1.2210848).
- Lowell, J., and G. Paton, 2018, Application of deep learning for seismic horizon interpretation: 88th Annual International Meeting, SEG, Expanded Abstracts, 1976–1980, doi: [10.1190/segam2018-2998176.1](https://doi.org/10.1190/segam2018-2998176.1).
- Marfurt, K. J., R. L. Kirlin, S. L. Farmer, and M. S. Bahorich, 1998, 3-D seismic attributes using a semblance-based coherency algorithm: *Geophysics*, **63**, 1150–1165, doi: [10.1190/1.1444415](https://doi.org/10.1190/1.1444415).
- Marfurt, K. J., V. Sudhaker, A. Gerszenkorn, K. D. Crawford, and S. E. Nissen, 1999, Coherency calculations in the presence of structural dip: *Geophysics*, **64**, 104–111, doi: [10.1190/1.1444508](https://doi.org/10.1190/1.1444508).
- Monniron, M., S. Frambat, S. Quillón, Y. Berthoumieu, and M. Donias, 2016, Seismic horizon and pseudo-geological time cube extraction based on a riemannian geodesic search: IEEE 12th Image, Video, and Multi-dimensional Signal Processing Workshop (IVMSP), 1–5.
- Parks, D., 2010, Seismic image flattening as a linear inverse problem: Master's thesis, Colorado School of Mines.
- Pedersen, S. I., T. Randen, L. Sonnenland, and Ø. Steen, 2002, Automatic fault extraction using artificial ants: 72nd Annual International Meeting, SEG, Expanded Abstracts, 512–515, doi: [10.1190/1.1817297](https://doi.org/10.1190/1.1817297).
- Pedersen, S. I., T. Skov, A. Hetlelid, P. Fayemendy, T. Randen, and L. Sonnenland, 2003, New paradigm of fault interpretation: 73rd Annual International Meeting, SEG, Expanded Abstracts, 350–353, doi: [10.1190/1.1817918](https://doi.org/10.1190/1.1817918).
- Pham, N., S. Fomel, and D. Dunlap, 2019, Automatic channel detection using deep learning: *Interpretation*, **7**, no. 3, SE43–SE50, doi: [10.1190/INT-2018-0202.1](https://doi.org/10.1190/INT-2018-0202.1).
- Posamentier, H., R. Davies, J. Cartwright, and L. Wood, 2007, Seismic geomorphology — An overview, in R. J. Davies, H. W. Posamentier, L. J. Wood, and J. A. Cartwright, eds., *Seismic geomorphology: Geological Society of London Special Publication*, 1–14.
- Qi, J., F. Li, and K. Marfurt, 2017, Multiazimuth coherence: *Geophysics*, **82**, no. 6, O83–O89, doi: [10.1190/geo2017-0196.1](https://doi.org/10.1190/geo2017-0196.1).

- Ren, S., K. He, R. Girshick, and J. Sun, 2015, Faster R-CNN: Towards real-time object detection with region proposal networks: Advances in Neural Information Processing Systems, 91–99.
- Rivenæs, J. C., C. Otterlei, E. Zachariassen, C. Dart, and J. Sjøholm, 2005, A 3D stochastic model integrating depth, fault and property uncertainty for planning robust wells, Njord Field, offshore Norway: Petroleum Geoscience, **11**, 57–65, doi: [10.1144/1354-079303-612](https://doi.org/10.1144/1354-079303-612).
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 234–241.
- Sandwell, D. T., 1987, Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data: Geophysical Research Letters, **14**, 139–142, doi: [10.1029/GL014i002p00139](https://doi.org/10.1029/GL014i002p00139).
- Shi, Y., X. Wu, and S. Fomel, 2019, Saltseg: Automatic 3D salt segmentation using a deep convolutional neural network: Interpretation, **7**, no. 3, SE113–SE122, doi: [10.1190/INT-2018-0235.1](https://doi.org/10.1190/INT-2018-0235.1).
- Stark, T. J., 2003, Unwrapping instantaneous phase to generate a relative geologic time volume: 73rd Annual International Meeting, SEG, Expanded Abstracts, 1707–1710, doi: [10.1190/1.1844072](https://doi.org/10.1190/1.1844072).
- Stark, T. J., 2004, Relative geologic time (age) volumes — Relating every seismic sample to a geologically reasonable horizon: The Leading Edge, **23**, 928–932, doi: [10.1190/1.1803505](https://doi.org/10.1190/1.1803505).
- Vail, P. R., R. G. Todd, and J. B. Sangree, 1977, Seismic stratigraphy and global changes of sea level — Part 5: Chronostratigraphic significance of seismic reflections: Section 2. Application of seismic reflection configuration to stratigraphic interpretation, in C. E. Payton, ed., Seismic stratigraphy — Applications to hydrocarbon exploration: AAPG Memoir 26, 99–116.
- Walsh, J., and J. Watterson, 1987, Distributions of cumulative displacement and seismic slip on a single normal fault surface: Journal of Structural Geology, **9**, 1039–1046, doi: [10.1016/0191-8141\(87\)90012-5](https://doi.org/10.1016/0191-8141(87)90012-5).
- Walsh, J. J., and J. Watterson, 1989, Displacement gradients on fault surfaces: Journal of Structural Geology, **11**, 307–316, doi: [10.1016/0191-8141\(89\)90070-9](https://doi.org/10.1016/0191-8141(89)90070-9).
- Wu, H., and B. Zhang, 2018, A deep convolutional encoder-decoder neural network in assisting seismic horizon tracking: arXiv preprint, arXiv:1804.06814.
- Wu, X., 2017, Directional structure-tensor based coherence to detect seismic faults and channels: Geophysics, **82**, no. 2, A13–A17, doi: [10.1190/geo2016-0473.1](https://doi.org/10.1190/geo2016-0473.1).
- Wu, X., and S. Fomel, 2018a, Automatic fault interpretation with optimal surface voting: Geophysics, **83**, no. 5, O67–O82, doi: [10.1190/geo2018-0115.1](https://doi.org/10.1190/geo2018-0115.1).
- Wu, X., and S. Fomel, 2018b, Least-squares horizons with local slopes and multigrid correlations: Geophysics, **83**, no. 4, IM29–IM40, doi: [10.1190/geo2017-0830.1](https://doi.org/10.1190/geo2017-0830.1).
- Wu, X., and D. Hale, 2013, Extracting horizons and sequence boundaries from 3D seismic images: 83rd Annual International Meeting, SEG, Expanded Abstracts, 1440–1445, doi: [10.1190/segam2013-0296.1](https://doi.org/10.1190/segam2013-0296.1).
- Wu, X., and D. Hale, 2015, Horizon volumes with interpreted constraints: Geophysics, **80**, no. 2, IM21–IM33, doi: [10.1190/geo2014-0212.1](https://doi.org/10.1190/geo2014-0212.1).
- Wu, X., and D. Hale, 2016, 3D seismic image processing for faults: Geophysics, **81**, no. 2, IM1–IM11, doi: [10.1190/geo2015-0380.1](https://doi.org/10.1190/geo2015-0380.1).
- Wu, X., L. Liang, Y. Shi, and S. Fomel, 2019, FaultSeg3D: Using synthetic datasets to train an end-to-end convolutional neural network for 3D seismic fault segmentation: Geophysics, **84**, no. 3, IM35–IM45, doi: [10.1190/geo2018-0646.1](https://doi.org/10.1190/geo2018-0646.1).
- Wu, X., Y. Shi, S. Fomel, and L. Liang, 2018, Convolutional neural networks for fault interpretation in seismic images: 88th Annual International Meeting, SEG, Expanded Abstracts, 1946–1950, doi: [10.1190/segam2018-2995341.1](https://doi.org/10.1190/segam2018-2995341.1).
- Wu, X., and G. Zhong, 2012, Generating a relative geologic time volume by 3D graph-cut phase unwrapping method with horizon and unconformity constraints: Geophysics, **77**, no. 4, O21–O34, doi: [10.1190/geo2011-0351.1](https://doi.org/10.1190/geo2011-0351.1).
- Wu, X., and Z. Zhu, 2017, Methods to enhance seismic faults and construct fault surfaces: Computers & Geosciences, **107**, 37–48, doi: [10.1016/j.cageo.2017.06.015](https://doi.org/10.1016/j.cageo.2017.06.015).
- Zeiler, M. D., and R. Fergus, 2014, Visualizing and understanding convolutional networks: European Conference on Computer Vision, Springer, 818–833.
- Zhao, T., and P. Mukhopadhyay, 2018, A fault-detection workflow using deep learning and image processing: 88th Annual International Meeting, SEG, Expanded Abstracts, 1966–1970, doi: [10.1190/segam2018-2997005.1](https://doi.org/10.1190/segam2018-2997005.1).
- Zinck, G., M. Donias, J. Daniel, S. Guillou, and O. Lavialle, 2013, Fast seismic horizon reconstruction based on local dip transformation: Journal of Applied Geophysics, **96**, 11–18, doi: [10.1016/j.jappgeo.2013.06.010](https://doi.org/10.1016/j.jappgeo.2013.06.010).