GEOPHYSICS, VOL. 86, NO. 2 (MARCH-APRIL 2021); P. IM51–IM62, 16 FIGS.
10.1190/GEO2020-0039.1

DOI:10.1190/geo2020-0039.1

Seismic horizon extraction with dynamic programming

Shangsheng Yan[1] and Xinming Wu[1]

ABSTRACT

Horizon picking is a fundamental and crucial step for seismic interpretation, but it remains a time-consuming task. Although various automatic methods have been developed to extract horizons in seismic images, most of them may fail to pick horizons across discontinuities such as faults and noise. To obtain more accurate horizons, we have developed a dynamic programming algorithm to efficiently refine manually or automatically extracted horizons so that they can more accurately track reflectors across discontinuities, follow consistent phases, and reveal more geologic details. In this method, we first compute an initial horizon using an automatic method, manual picking, or interpolation with several control points. The initial horizon may not be accurate, and it only needs to follow the general trend of the target horizon. Then, we extract a subvolume of amplitudes centered at the initial horizon and meanwhile flatten the subvolume according to the initial horizon. Finally, we use the dynamic programming to efficiently pick the globally optimal path that passes through the global maximum or minimum amplitudes in the subvolume. As a result, we are able to refine the initial horizon to a more accurate horizon that follows consistent amplitude peaks, troughs, or zero crossings. Because our method does not strictly depend on the initial horizon, we prefer to directly interpolate an initial horizon from a limited number of control points, which is computationally more efficient than automatically or manually picking an initial horizon. In addition, our method is convenient to interactively implement to update the horizon while editing or moving the control points. More importantly, these control points are not required to be exactly placed on the target horizon, which makes the human interaction highly convenient and efficient. We develop our method with multiple 2D and 3D field examples that are complicated by noise, faults, and salt bodies.

INTRODUCTION

Extracting horizons from seismic images is one of the fundamental steps for geophysical interpretation. Seismic horizons are considered to be matched with stratal surfaces of constant geologic time, which represent geologically synchronous surfaces (Vail et al., 1977). Therefore, horizons can be used to construct a chronostratigraphic model (e.g., Labrunye et al., 2009; Labrunye and Carn, 2015). They are important for identifying structural and stratigraphic features (Qayyum et al., 2018) because faults (e.g., Wang and AlRegib, 2014; Wang et al., 2014), channels (Hale, 2009), unconformities (Bugge et al., 2018), and the boundaries of salt bodies (e.g., Shafiq et al., 2015; Wang et al., 2015c; Shafiq and AlRegib, 2016) can be indicated by discontinuities or terminations of horizons. They can be used to analyze ancient depositional environments and geomorphic features (Posamentier et al., 2007). Besides, high-density sets of horizons, usually called horizon cubes (de Groot et al., 2010), are also significant for generating the Wheeler diagram (e.g., Qayyum et al., 2015, 2017), which is one of the key methodologies for sequence stratigraphic interpretation (Qayyum et al., 2012).

Manually picking seismic horizons is time-consuming because seismic images can be huge (especially in 3D) and complex. To solve this problem, many types of automatic or semiautomatic methods have been developed to more efficiently extract horizons. One type of method is based on waveform similarities (e.g., Borgos et al., 2003; Lacaze and Valding, 2009), in which they use the coherence or correlation between neighboring traces to track horizons. These methods often follow small variations in structure and are locally optimal, and therefore are effective in revealing detailed structures. However, these methods may fail to extract consistent horizons while passing through some discontinuities, including noise and faults (Dorn, 2011).

Manuscript received by the Editor 28 January 2020; revised manuscript received 23 November 2020; published ahead of production 13 December 2020; published online 11 February 2021.
[1]University of Science and Technology of China, School of Earth and Space Sciences, Hefei 230026, China. E-mail: ssyan33@mail.ustc.edu.cn; xinmwu@ustc.edu.cn (corresponding author)

DOI:10.1190/geo2020-0039.1

Another type of method is to compute the relative geologic time (RGT) values with some techniques and then extract the RGT contours to obtain horizons (e.g., Stark, 2005; Wu and Zhong, 2012). These methods are globally optimal but often generate smooth horizons that lack detailed geologic information. In addition, significant human interaction is required to deal with seismic discontinuities including faults and unconformities.

The most commonly used type is the slope-based methods (e.g., Fomel, 2010; Parks, 2010; Di et al., 2018; Lou et al., 2019) or normal vectors (Luo and Hale, 2013) of seismic reflectors. Seismic reflection slopes can also be obtained through computing the structure tensor (e.g., Bakker et al., 1999; Bakker, 2002; Fehmers and Höcker, 2003; Hale, 2009; Morelatto and Biloti, 2013) or using the plane-wave destruction method (e.g., Fomel, 2002; Phillips and Fomel, 2016). These slope-based methods are indeed robust for noise, but they fail to reveal detailed geologic structures due to averaging the local features. Some other slope-based methods, using semblance scanning (e.g., Marfurt et al., 1998; Marfurt, 2006), dynamic image warping (e.g., Hale, 2013; Arias, 2016), and 2D log-Gabor filtering (Yu et al., 2013), are effective only when reflectors are laterally continuous.

Many slope-based methods cannot pick correct horizons across a fault (e.g., Lomask et al., 2006; Parks, 2010; Wu and Hale, 2013, 2015; Zinck et al., 2013; Monniron et al., 2016) because slopes cannot follow correct reflectors across faults. To handle this issue, some methods remove the faults from a seismic image (e.g., Luo and Hale, 2013; Wu et al., 2016) before extracting horizons or they extract horizons by using manual control points (often placed on the opposite sides of a fault) as constraints (Wu and Hale, 2015). Some methods are proposed to extract horizons without removing faults, where computing multigrid correlations (Wu and Fomel, 2018) and nonlocal matching of seismic traces (Bugge et al., 2019) help track the corresponding reflections passing through a fault. Some other methods formulate horizon extraction as the problem of solving an inhomogeneous anisotropic Poisson equation with faults as interior boundaries (e.g., Wang et al., 2015a, 2015b).

In our study, we propose a method with the dynamic programming algorithm (e.g., Bellman and Dreyfus, 1962; Cormen et al., 2001) to effectively and efficiently extract horizons from seismic images. In our method, we start with an initial horizon that can be manually interpreted, automatically picked, or interpolated from several control points. Such an initial horizon is often not accurate enough to follow consistent phases (peaks, troughs, or zero crossings) or consistent reflections across discontinuities (faults or noise). We then refine or correct the horizon by using the dynamic programming to search for the globally optimal path (two dimensions) or surface (three dimensions) in a depth or time window centered at the initial horizon. In the following sections, we first introduce the conventional horizon extraction methods, including manual interpretation and the automatic algorithm using local slopes. Meanwhile, we display some 2D and 3D data to illustrate the potential problems of these methods. Then, we describe how dynamic programming optimizes the result in these data. In addition, we discuss the influence of the number and location of control points on our method. Finally, to demonstrate the feasibility of our method, we show more 3D applications and compare the results between three methods with only slopes, local slopes and multigrid correlations, and our proposed method, respectively.

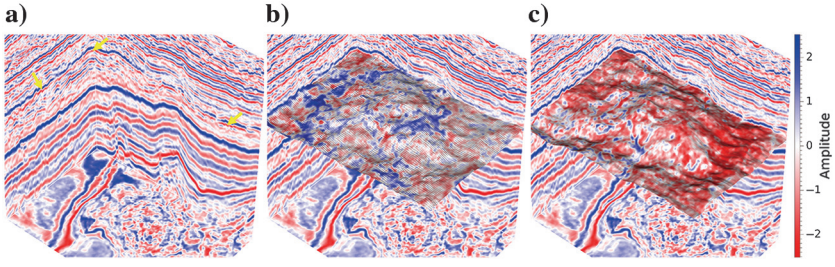

Figure 1. (a) A 3D seismic image from the publically available Volve data, where the yellow arrows denote the target horizon with the troughs (the red color). (b) A manually interpreted horizon provided along with the seismic data set. (c) The refined horizon updated by our method, which uses (b) as an initialization. The extracted surface in (c) with the most red color suggests that our method can follow more consistent phases than manual picking (b).

CONVENTIONAL HORIZON EXTRACTION

Manually picking horizons not only requires a lot of time and effort, but it is also highly subjective due to human factors. Therefore, different automatic methods have been developed, which can be divided into three categories: amplitude-based methods, slope-based methods, and methods with RGT. Slope-based methods are ones most commonly used to extract horizons by following reflection slopes, which are also used in this study as comparisons.

Manual horizon extraction

Manually interpreting seismic horizons is highly time-consuming and labor-intensive, especially in 3D. To save time, interpreters often sparsely interpret only a limited number of inline and crossline sections on a coarse grid and then pick the rest of the sections by tracking the consistent amplitudes. However, these methods are usually rough and only follow local changes. In addition, it is highly subjective when passing through noise and discontinuities. A common method is based on autocorrelation to interpolate the picked lines, which tracks amplitude changes trace by trace. In general, the accuracy and resolution of manually interpreted horizons are often limited.

Figure 1b shows a horizon surface that is manually interpreted in a subset (500 [inline] × 330 [crossline] × 300 [depth] samples) of the 3D Volve seismic data set shown in Figure 1a. The Volve data set is acquired in the central part of the North Sea, and it is kindly made public by Equinor (2018).

The yellow arrows denote the horizon of troughs that we expect to extract. We observe that the manually interpreted horizon surface (provided along with the seismic data) does not follow a consistent phase and jumps between the peaks and troughs frequently because the color (denotes amplitude) variation is obvious as shown on the surface. Besides, some artifacts of square patterns are observed on the surface, which is probably due to the interpolation from sparsely interpreted inline and crossline slices. Such interpolation typically blurs structures and reduces accuracy. To improve or refine the manually interpreted horizon, we use the proposed

method to further update the horizon so that it more consistently follows amplitude troughs as shown in Figure 1c.

Automatic method with slopes/dips

For the conventional automatic methods, we mainly introduce the commonly used slope-based method in this study. The slope can be estimated by various methods, such as plane-wave destruction (Fomel, 2002), structure tensors (e.g., Bakker et al., 1999; Fehmers and Höcker, 2003; Hale, 2009; Morelatto and Biloti, 2013), and semblance scanning (e.g., Marfurt et al., 1998; Marfurt, 2006).

In this study, we first estimate seismic normal vectors $\mathbf{u}(\mathbf{x})$ using structure tensors (e.g., Hale, 2009; Wu and Fomel, 2018), which are perpendicular to seismic reflections. The normal vectors are unit vectors containing the vertical, inline, and crossline components, denoted by $u_1(\mathbf{x})$, $u_2(\mathbf{x})$, and $u_3(\mathbf{x})$, respectively. Then, we use the following equations to calculate the inline and crossline slopes:

$$p(\mathbf{x}) = -\frac{u_2(x)}{u_1(\mathbf{x})} \quad \text{and} \quad (\mathbf{x}) = -\frac{u_3(x)}{u_1(\mathbf{x})}. \tag{1}$$

Then, we can find a target horizon $z(x)$ using the slope-based method by matching horizon slopes and the preestimated reflection slopes with the least-squares method (e.g., Parks, 2010; Wu and Hale, 2013, 2015):

$$\begin{bmatrix} w(x,z(x))\frac{\partial z(x)}{\partial x} \\ \mu\frac{\partial^2 z(x)}{\partial x^2} \end{bmatrix} \approx \begin{bmatrix} w(x,z(x))p(x,z(x)) \\ 0 \end{bmatrix}, \tag{2}$$

where $w(x,z)$ is used to measure the quality of the computed reflection slopes, which satisfies $0 \le w(x,z) \le 1$. The second equation is used to smooth the target horizon $z(x)$ as a regularization constraint. The small constant μ is to balance the regularization and slope-based equations, for example, $\mu = 0.001$.

We can readily extend equation 2 to 3D seismic images to extract a horizon surface $z(x,y)$:

$$\begin{bmatrix} w(x,y,z)\frac{\partial z}{\partial x} \\ w(x,y,z)\frac{\partial z}{\partial y} \\ \mu\left(\frac{\partial^2 z}{\partial x^2}+\frac{\partial^2 z}{\partial y^2}\right) \end{bmatrix} \approx \begin{bmatrix} w(x,y,z)p(x,y,z) \\ w(x,y,z)q(x,y,z) \\ 0 \end{bmatrix}, \tag{3}$$

where $p(x,y,z)$ and $q(x,y,z)$ denote the preestimated inline and crossline reflection slopes at the horizon surface $z(x,y)$, respectively.

In equations 2 and 3, the right side slopes p and q depend on the horizon $z(x)$ (two dimensions) or $z(x,y)$ (three dimensions) to be solved, which indicates that these equations are nonlinear. Therefore, these slope-based methods need to start with an initial horizon and then iteratively refine the horizon until the slopes of the target horizon fit the preestimated slopes at the positions of the horizon. The initial horizon is typically interpolated from some predefined control points. If there is only one control point, then the initial horizon is a horizontal line (two dimensions) or plane (three dimensions) passing through the control point.

Figure 2b shows a slope-based horizon that is extracted by solving equation 1 with one control point (the red circle). This horizon follows the general structure trend, but it fails to track consistent phases. It jumps from peaks to troughs (denoted by the yellow arrow in Figure 2b) when it goes across the small fault. To address this problem, we propose a new method with the dynamic programming algorithm to update or refine the horizon in Figure 2c, in which the refined horizon more consistently follows the amplitude peaks.

We further display a 3D example in Figures 3 and 4, in which the horizons are displayed by amplitude and time, respectively. The yellow arrows (indicate the troughs) in both figures denote the target horizon that is supposed to be picked properly. The horizon surface (Figure 3b) extracted by the slope-based method (equation 2) jumps between peaks and troughs because we can observe obvious color variations on the surface. In addition, this horizon surface is smooth and therefore the geologic details on the surface are blurred. The horizon refined by our method in Figure 3c is colored blue almost everywhere, which displays more consistent phase and detailed structures, such as channels denoted by the white arrow. In addition, the small structures and channels are readily apparent in the time-colored horizon (Figure 4c).

Because the above automatic method with only slopes often cannot accurately track consistent horizons passing through discontinuous areas as displayed in Figures 2b and 3b, Wu and Fomel (2018) improve the method by taking into account the spatially global constraint. In this method, coarse grid correlations are introduced to additionally constrain equations 2 and 3. This constraint is helpful for tracking horizons across discontinuities. More detailed information can be found in Wu and Fomel (2018). In our study, we will use these two methods (the one with only slopes and the one with the slope and multigrid correlations) as benchmarks to demonstrate the performance of our proposed method based on dynamic programming. However, our method is not based on these previously published methods.

HORIZON REFINEMENT

To extract more accurate horizons as shown in Figures 1c, 2c, 3c, and 4b, we propose an efficient horizon refinement method with dynamic programming (e.g., Bellman and Dreyfus, 1962; Cormen et al., 2001; Hale, 2013). In this method, we first start with an initial horizon that can be obtained by manual interpretation, the slope-based methods, or interpolation algorithms. Then, we update it to a more accurate horizon with dynamic programming.

2D horizon refinement

In the extracting horizons problem, we can assume a horizon consistently passes amplitude peaks, troughs, or zero crossings.

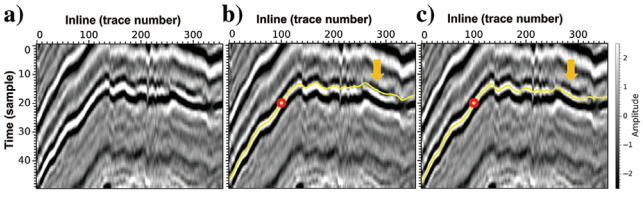

Figure 2. (a) In a 2D seismic image, one horizon (the yellow curve) is first extracted by (b) a slope-based method and further improved or refined by using (c) the proposed dynamic programming method. The area denoted by the yellow arrows shows that our method significantly improves the horizon so that it follows consistent troughs. The red circle is a control point used in the horizon extraction.

Therefore, we formulate our task of tracking a seismic horizon as a problem of finding the globally minimum or maximum path (two dimensions) or surface (three dimensions) in a seismic amplitude image and propose to solve the problem by using a modified dynamic programming algorithm. Dynamic programming is widely used to effectively and efficiently solve for global optimization problems (Cormen et al., 2001).

Figure 5 shows our proposed workflow of using the dynamic programming algorithm to refine a 2D horizon. In this workflow, we start with an initial horizon (the yellow curve in Figure 5a) that is automatically computed with one control point (the red circle in Figure 5a) by using the slope-based method (equation 2). This initial horizon is close to the target horizon of amplitude troughs (colored in black in the background seismic image), but it does accurately follow the troughs and therefore is required to be improved or refined, especially for the right part of the horizon. Such an initial horizon in our refinement workflow only needs to be close to the target horizon and therefore can be a rough interpretation that can also be easily obtained by interpolation or manual interpretation with a limited number of control points. With such an initial horizon, we then extract a small seismic image section in a window that is vertically centered at the initial horizon. To improve the computational efficiency, such a vertical window (denoted by the vertical red lines in Figure 5a) is typically chosen be as narrow as necessary but should include the target horizon to be extracted. Figure 5b shows the seismic section extracted in the vertical window (with 21 samples in total in the vertical direction) and flattened according to the initial horizon.

In this flattened seismic section (Figure 5b), the yellow arrows denote the target horizon. The target horizon can be considered as the minimum path passing through most amplitude troughs (colored in black in Figure 5b) from left to right. Therefore, we extract the target horizon by solving the following global minimization problem:

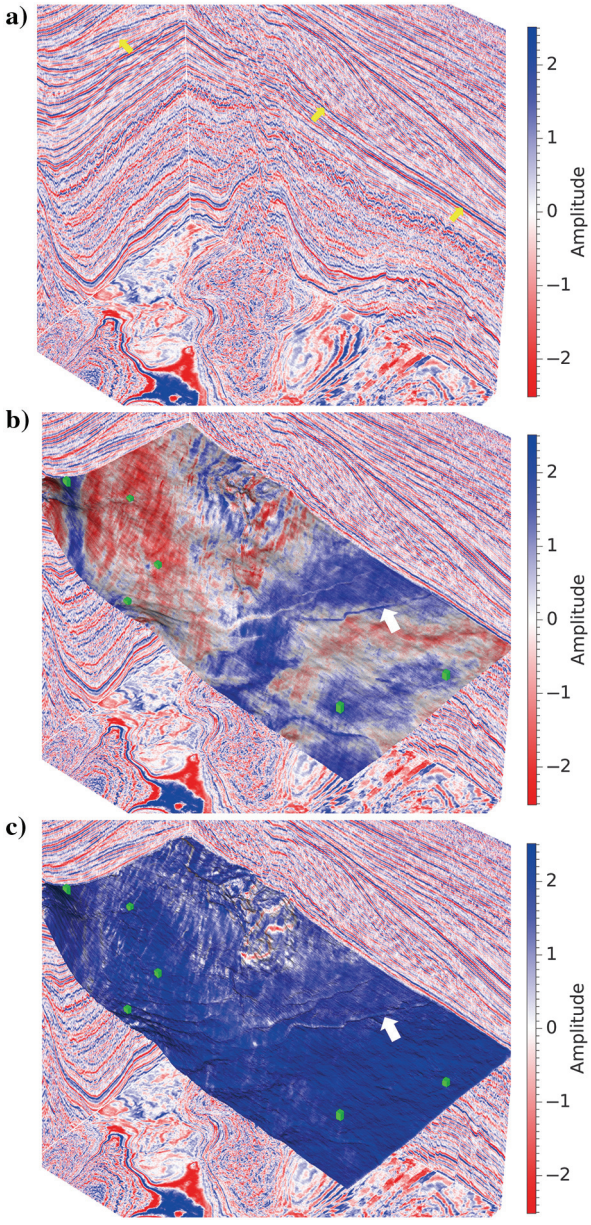

Figure 3. (a) In a 3D seismic image, the horizon denoted by the yellow arrows (the blue color) is extracted by (b) the slope-based method and (c) our method, respectively. The extracted horizon in (c) follows more consistent phases than that in (b) because the horizon is colored blue almost everywhere in (c). Besides, the result in (c) reveals detailed structures, such as small channels denoted by the white arrow. The green squares are control points used in horizon extraction.

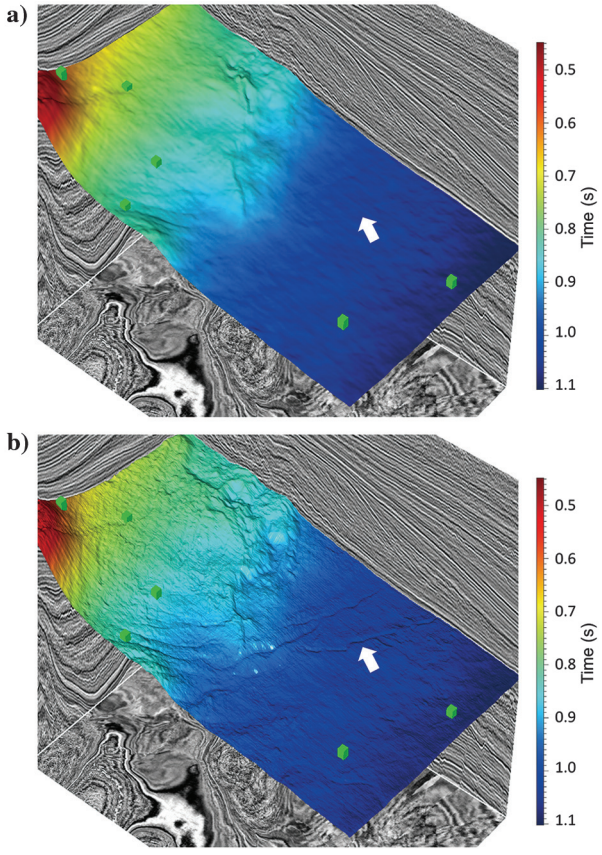

Figure 4. The horizons in Figure 3b and 3c are displayed with time in (a and b), respectively. Similarly, we are able to observe much more detailed structures including the channels on the horizon by (b) our proposed method than the one by (a) the slope-based method.

DOI:10.1190/geo2020-0039.1

$$E1 = \arg\min_{j[i]} \sum_{i=1}^{N} g[i, j[i]],$$
$$\text{s.t.} \left| j[i+1] - j[i] \right| \le \varepsilon (0 < \varepsilon < 1), \tag{4}$$

where i and j represent the horizontal and vertical axes, respectively, $j[i]$ indicates a path from left to right, and $g[i, j]$ is defined as a function of the seismic amplitude image $a[i, j]$ as follows:

$$g[i,j] = \begin{cases} a[i,j], & \text{pick a horizon of troughs} \\ -a[i,j], & \text{pick a horizon of peaks} \\ |a[i,j]|, & \text{pick a horizon of zero crossings} \end{cases}. \tag{5}$$

With such a definition, we make sure that the target horizon of troughs, peaks, or zero crossings is a minimum path in the input image $g[i, j]$.

The second part of the objective function (equation 4) represents the slope constraint imposed on the path to be computed. In solving this constrained minimization problem, we aim to find the target horizon as the global minimum path such that the total summation of the amplitude values on the path is minimized. In the flattened seismic image, the target horizon should be smooth and almost flat; therefore, we typically use a small slope constraint (e.g., $\varepsilon = 0.25$) in finding the optimal path. To solve this constrained nonlinear minimization problem, we use a modified dynamic programming algorithm (Hale, 2013) with three steps of nonlinear smoothing, forward accumulation, and backward tracking.

Nonlinear smoothing

As shown in Figure 5b, the minimum path to be picked in the input image $g[i, j]$ is not obvious and many local low-amplitude samples do not correspond to the target horizon or the minimum path in the middle (denoted by the yellow arrow in Figure 5b). Picking the minimum path is still not trivial. To further highlight the minimal path, we apply a nonlinear smoothing processing (Hale, 2013) to enhance the low amplitudes corresponding to the minimal path but eliminate the locally low amplitudes that are unrelated to the path. The nonlinear smoothing is a combination of forward and backward nonlinear accumulation applied to the input image $g[i, j]$. The forward accumulation with a slope constraint $|j[i+1] - j[i]| \le \varepsilon$ is expressed as shown in equation 6:

$$f[0, j] = g[0, j],$$
$$f[i,j] = g[i,j] + \min \begin{cases} f[i-d, j-1] + \sum_{k=i-d+1}^{i-1} g[k, j-1] \\ f[i-1, j] \\ f[i-d, j+1] + \sum_{k=i-d+1}^{i-1} g[k, j+1] \end{cases}$$
$$\text{for } i = 1, 2, \ldots, N-1, \tag{6}$$

where $f[i, j]$ is the output by accumulating the minimum value in a local window at every sample. The term d is the integer that is closest to $1/\varepsilon$, which defines the lateral window in searching for the locally minimum accumulation at every sample. Similarly, the backward accumulation with the constraint $|j[i-1] - j[i]| \le \varepsilon$ is expressed as shown in equation 7:

$$b[N-1, j] = g[N-1, j],$$
$$b[i,j] = g[i,j] + \min \begin{cases} b[i+d, j-1] + \sum_{k=i+1}^{i+d-1} g[k, j-1] \\ b[i+1, j] \\ b[i+d, j+1] + \sum_{k=i+1}^{i+d-1} g[k, j+1] \end{cases}$$
$$\text{for } i = N-2, N-3, \ldots, 0. \tag{7}$$

Either the forward or backward accumulations can be regarded as the smoothness of $g[i, j]$ in the forward and backward directions, respectively.

By combining the forward and backward accumulations of the input image, we obtain a two-sided nonlinear smoothing result as follows:

$$s[i, j] = f[i, j] + b[i, j] - g[i, j], \tag{8}$$

where the input image $g[i, j]$ is subtracted because it has been added in the forward and reverse accumulations. After the nonlinear smoothing in Figure 5b, we obtain the smoothed output (Figure 5c) where the minimum path (or target horizon) is significantly enhanced, whereas the locally low amplitudes away from the path are suppressed. Therefore, picking the minimum path in Figure 5c is much more straightforward than in Figure 5b.

Forward accumulation and backtracking

To pick the minimum path from the smoothed image $s[i, j]$, we apply another forward accumulation to $s[i, j]$ to obtain an accumulated image $c[i, j]$ (the background image in Figure 5d) from which we then backtrack the minimum path $j[i]$ as follows:

$$j[N-1] = \arg\min_{j} c[N-1, j],$$
$$l = j[i],$$
$$j[i-1] = \arg\min_{l-1, l, l+1} \begin{cases} c[i-d, l-1] + \sum_{k=i-d+1}^{i-1} c[k, l-1] \\ c[i-1, l] \\ c[i-d, l+1] + \sum_{k=i-d+1}^{i-1} c[k, l+1] \end{cases}$$
$$\text{for } i = N-1, N-2, \ldots, 1. \tag{9}$$

The green curve in Figure 5d shows the picked minimum path. Furthermore, we map the picked minimum path from the flattened space back to the original space and obtain the final horizon as shown as the green curve in Figure 5e. By applying this 2D horizon refinement workflow, we effectively and efficiently update the initial horizon (the yellow curve in Figure 5e) to more accurately follow the nearby target horizon. Figure 2c shows another example of using the workflow to effectively refine a horizon of amplitude peaks.

3D horizon surface refinement

As the dynamic programming is inherently a 2D algorithm, we need some modifications to make it applicable to extract the minimum surface for 3D horizon surface refinement. Figure 6 shows the workflow of using dynamic programming in refining a 3D seismic horizon surface. Similar to the 2D horizon refinement, we again

start with an initial horizon surface (Figure 6a), which is close to the nearby target horizon but is not accurate enough because we can still observe significant amplitude or color variations on the surface.

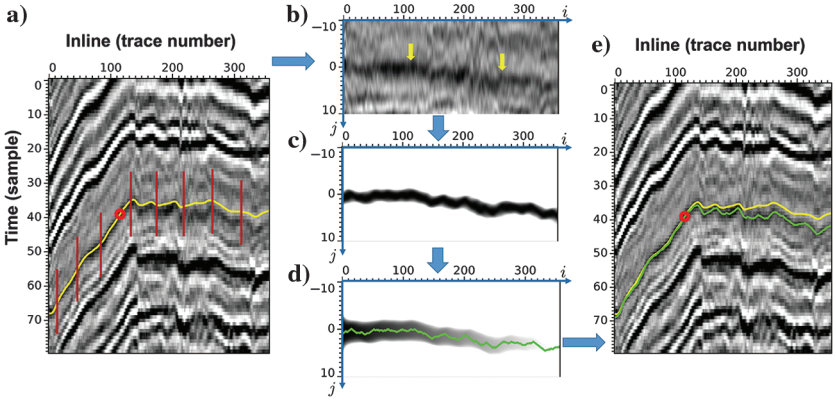

Figure 5. The proposed workflow of refining a horizon in a 2D seismic image. Step 1: (a) We first compute an initial horizon (the yellow curve) with a control point (the red circle), and then we choose a window (denoted by the red lines) vertically centered at the initial horizon. Step 2: (b) We extract a subimage from the window (vertically with 21 samples) and flatten it according to the centered initial horizon. Step 3: (c) We apply a nonlinear smoothing to the flattened subimage to enhance the minimum path or target horizon to be picked. Step 4: (d) We apply a forward accumulation to the smoothed image and backtrack the minimum path (denoted by the green curve). Step 5: (e) We further map the picked minimum path from the flattened space back to the original space to obtain the finally refined horizon (the green curve), which more accurately follows the target horizon than the initial horizon (the yellow curve).

Figure 6b shows the 3D seismic amplitude image flattened according to the initial horizon surface. In this flattened seismic image, the target horizon is considered as the surface passing through minimum seismic amplitudes as colored by the red in the middle of the image. Therefore, we pick the target horizon $j[i,k]$ by solving the following constrained minimization problem:

$$E2 = \arg\min_{j[i,k]} \sum_{i=1}^{N}\sum_{k=1}^{M} g[i,k,j[i,k]],$$
$$\text{s.t.} \left|j[i+1,k] - j[i,k]\right| \le \varepsilon$$
$$\text{and} \left|j[i,k+1] - j[i,k]\right| \le \varepsilon, \tag{10}$$

where $g[i,k,j]$ is the input amplitude image as defined in equation 5. The term ε is a constant number $(0 \le \varepsilon \le 1)$, which provides the slope constraints (in the i-axis and k-axis directions) in picking the minimum surface $j[i,k]$. In the flattened seismic image (flattened according to the initial horizon), the target horizon should be smooth and almost flat because the initial horizon is typically close to the target horizon. Therefore, we typically define a small slope constraint (e.g., $\varepsilon = 0.25$) to pick a smooth surface with small slopes in the i-axis and k-axis directions.

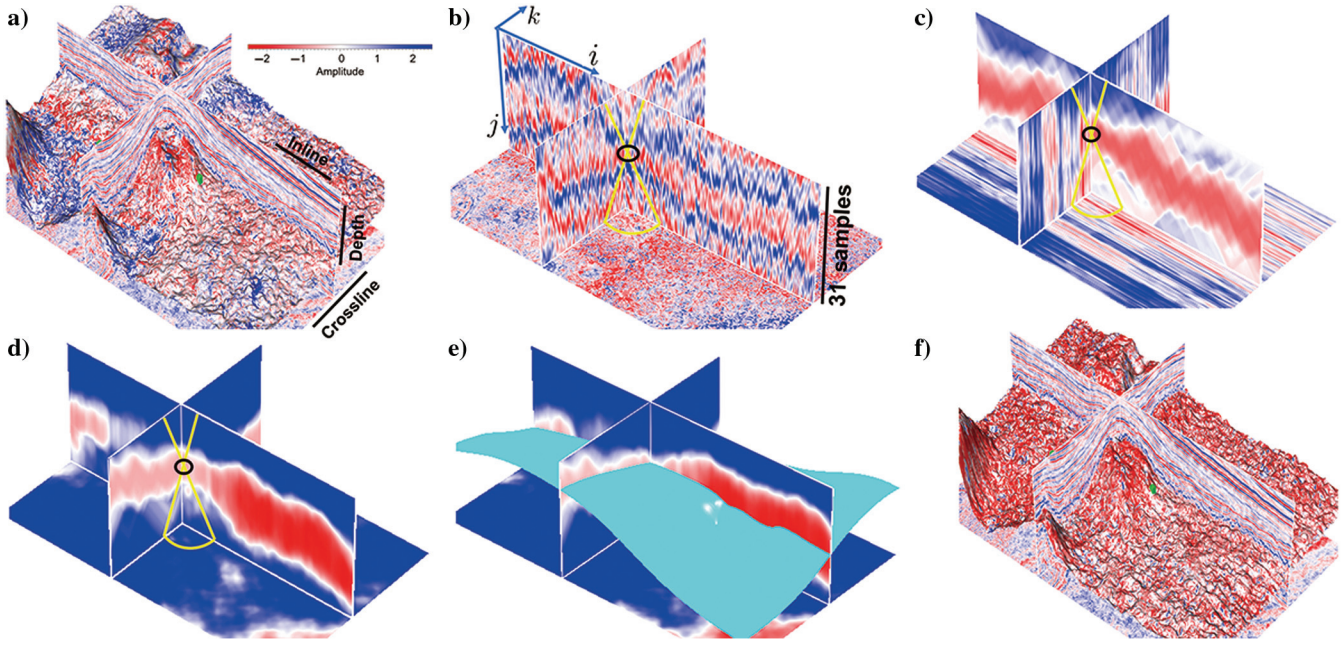

Figure 6. The workflow of refining a horizon surface in a 3D seismic image, which is similar to the 2D workflow. Step 1: (a) We first compute an initial horizon with two control points (the green squares). Step 2: (b) We choose a window centered at the initial horizon and extract a subimage volume from the window (vertically with 31 samples) and then flatten it according to the initial horizon surface. Step 3: (c) We apply the 2D nonlinear smoothing in all i-j sections to enhance the target minimal surface. Step 4: (d) We further apply the 2D nonlinear smoothing in all i-k sections to obtain a finally smoothed image. Step 5: (e) We apply a forward accumulation to the smoothed image and backtrack the minimum path (denoted by the cyan surface). Step 6: (f) We finally map the picked minimal surface from the flattened space back to the original space to obtain the finally refined horizon surface (the red surface). In addition, if we want to force the target horizon to pass through a given control point (the black circles in [b-d]), the maximums are set away from the control point and the minimums near the point in the surrounded area by the yellow lines.

Automatically picking the target minimum surface, directly from the flattened seismic image (Figure 6b), is still not straightforward because of the noise, discontinuities, and local minimum amplitudes around the target surface in the seismic image. Similar to the 2D minimal path picking, we apply the 2D nonlinear smoothing (equation 8) to the 3D seismic image in all $i-j$ sections and obtain a smoothed image in Figure 6c in which the target minimal surface is significantly enhanced, especially in the $i-j$ sections. We further apply the 2D nonlinear smoothing to all $k-j$ sections to obtain the finally smoothed image (Figure 6d) in which the target minimal surface is much more obvious than the one in the original seismic image (Figure 6b). From this smoothed image, we further pick the minimal surface as a set of 2D minimal paths by using a combination of forward accumulation (equation 6) and backtracking (equation 9). By applying nonlinear smoothing, we have imposed the slope constraints in the i-axis and k-axis directions to ensure that the minimal surface is smooth in both directions. Therefore, we can pick the 2D minimal paths in either the $i-j$ sections or the $k-j$ sections, which will form a consistent and smooth surface (Figure 6e). Finally, we map the picked minimal surface from the flattened space back to the original space to obtain the finally updated or refined horizon surface as shown in Figure 6f. The amplitudes colored on the refined horizon surface (Figure 6f) are more consistent than the initial horizon (Figure 6a), which indicates that the refined horizon is more accurate.

The results in Figures 1c and 3c demonstrate the effectiveness of the proposed workflow to refine a horizon surface interpreted by manual and automatic picking, respectively. To further analyze its performance, Figure 7 shows a 2D slice view extracted from Figure 3 in the inline-depth direction. In Figure 7b, the horizon obtained by the dynamic programming (the red curve) follows more consistent peaks than the one obtained by the slope-based method (the green curve).

MORE 3D APPLICATIONS

To further demonstrate the feasibility of the developed method, we verify our method in another 3D example (Figure 8) and we compare the results obtained by methods with only slopes (Fomel, 2010), local slopes and multigrid correlations (Wu and Fomel, 2018), and dynamic programming.

We apply our method to the same data used in Wu and Fomel (2018). Figure 8 indicates a 3D seismic image from the Netherlands offshore F3 block seismic data with size of (951,550,242) samples in the inline, crossline, and depth directions. This image contains complex structures, such as a large fault (white line) and salt bodies. The yellow and black arrows in Figure 8, respectively, denote the first and second horizons we want to extract.

In Figure 9, we use the three methods to pick the horizon that is denoted by the yellow arrows in Figure 8 with two control points (denoted by the green squares). We expect the picked horizon to follow the amplitude troughs. Therefore, the horizon surface, colored by the amplitudes, is expected to display consistent red colors. The horizon obtained by the automatic method with only slopes (Figure 9a) shows an inconsistent phase, in which the inconsistent colors on the extracted horizon surface suggest that it jumps between peaks (blue) and troughs (red). This is probably because the horizon passes through a lot of small faults but the local slopes fail to follow reflections dislocated by the faults. Figure 9b displays the horizon calculated by the method with local slopes and multigrid correlations. This indicates that this method improves the picked horizon in most areas. However, we still observe some phase or color variation near the large fault as denoted by the black arrow in Figure 9b. Figure 9c shows the refined horizon using dynamic programming with Figure 9a as the initialization. The picked horizon surface shows consistent troughs and exhibits a red color almost everywhere. In addition, the computational cost of this method is far less than the other two methods (Figure 11) because our proposed dynamic programming can use the interpolation to calculate the initial horizon to replace the slope-based step in the other two methods.

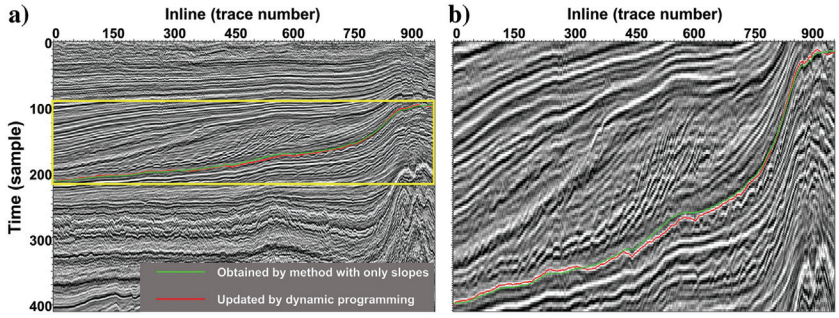

Figure 7. To view the local details of the extracted horizon surfaces in Figure 3, we extract a 2D inline seismic section from Figure 3 and show it in (a). The green and red curves denote where the horizon surfaces intersect with the seismic section. From the close view (b) of the yellow box in (a), we observe that the horizon (the red curve) refined by our method more consistently follows the amplitude peaks than the one (the green curve) computed by the slopes.

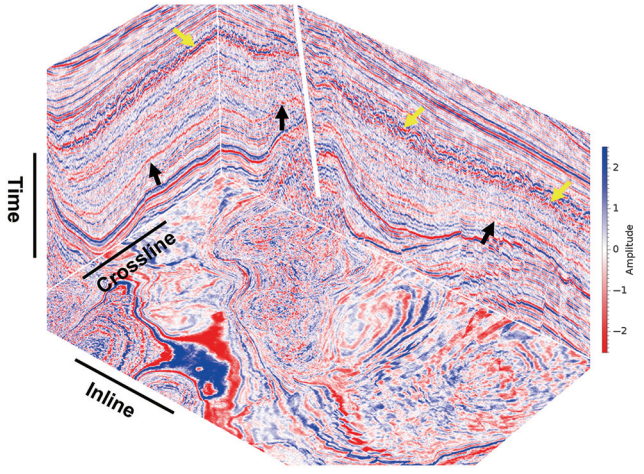

Figure 8. A 3D seismic image with many faults including a major fault (the white line) and some salt bodies. The first horizon with the yellow arrows (pointing to the troughs) is apparently discontinuous, which is complicated with a large fault and many small faults. The second target horizon, denoted by the black arrows (pointing to the troughs), is complicated by the large fault as well as salt bodies at the bottom.

In Figure 10, we select four control points (the green squares) to pick the horizon that is denoted by the black arrows in Figure 8. This horizon is more complex including a large fault (the white line) and some salt bodies as shown in the image. Figure 10a displays the extracted horizon by the method with only slopes. We select the control points on amplitude troughs, so the accurate horizon is supposed to be in red everywhere. However, the blue color on the picked surface suggests that the horizon cannot follow consistent phases and tracks incorrect positions to the peaks. Figure 10b shows the result by the method with local slopes and multigrid correlations, where the horizon is red almost everywhere. Similarly, the horizon obtained by the dynamic programming in Figure 10c exhibits consistent troughs as well. This indicates that both methods

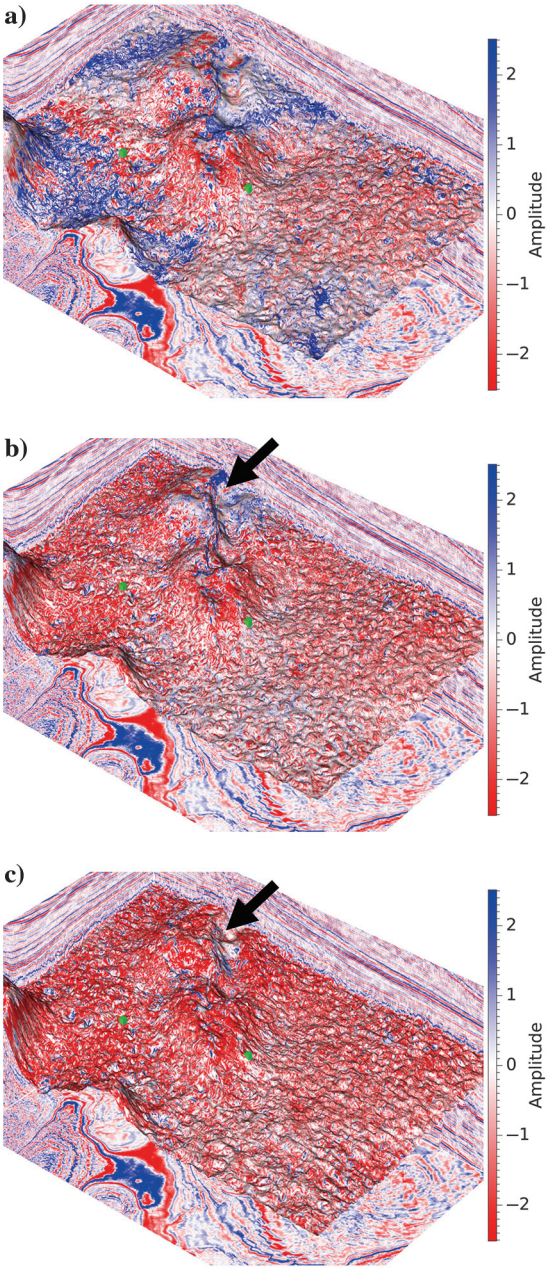

Figure 9. The first horizon from the 3D image (Figure 8) is extracted by methods with (a) only slopes, (b) local slopes and multigrid correlations, and (c) dynamic programming, respectively. For all three methods, we use two control points (the green squares). (c) The horizon surface obtained by the dynamic programming follows more consistent troughs than those in (a and b) because it is visually more consistently colored in red. In addition, the proposed dynamic programming method is much more efficient than the other two methods.

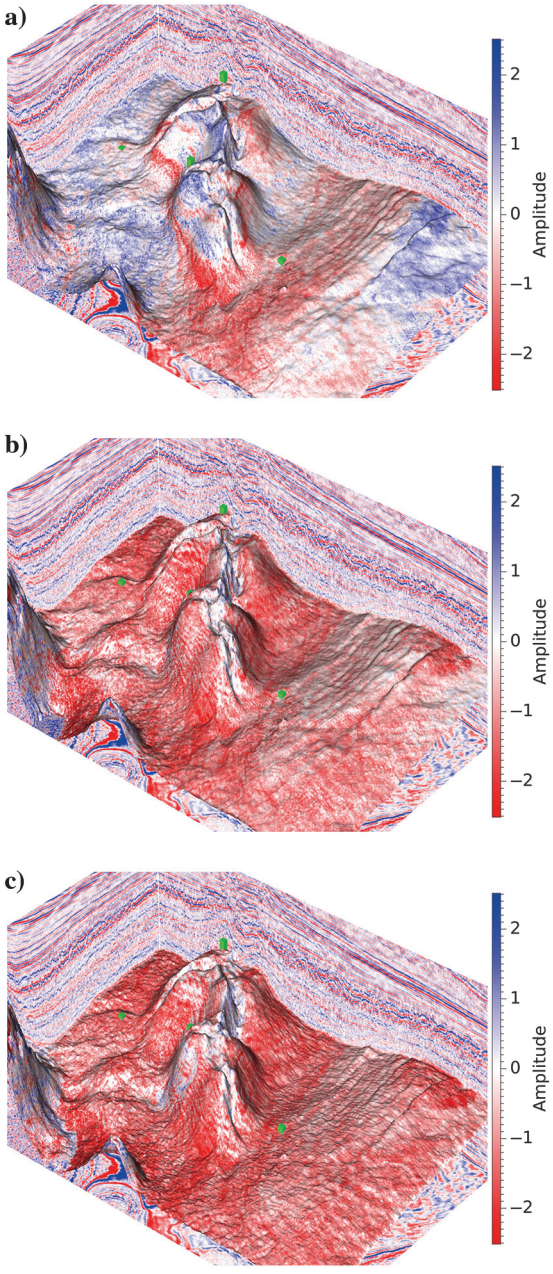

Figure 10. The second horizon from 3D image (Figure 8) is extracted by methods with (a) only slopes, (b) local slopes and multigrid correlations, and (c) the dynamic programming, respectively. We use four control points (the small green cubes) for all three methods. The refined horizon in (c) exhibits comparable performance in comparison with the horizon in (b) and is significantly improved compared to the one in (a).

in Figure 10b and 10c can obtain comparable results across faults and salt bodies.

Figure 11 shows a quantitative comparison of the computational costs of the methods in extracting the horizons (Figures 9 and 10). This table indicates that our method with dynamic programming runs two times faster than the slope-based method and six times faster than the multigrid method. These two methods need to iteratively solve large and nonlinear systems. However, in our proposed method, we use a simple and fast linear interpolation method to replace the above steps to obtain the initial horizons. Then, we update the horizons by dynamic programming, which can yield results comparable to the other two methods (as shown in Figures 9 and 10), but it takes much less time.

DISCUSSION

Initialization

Our proposed method contains two steps: (1) calculating an initial horizon and (2) refining the horizon with the dynamic programming algorithm. For the first step, a stable way to obtain an initial horizon is to use the result by the above slope-based method. However, this type of method required computing reflection slopes and iteratively solving for large linear systems (especially for huge 3D seismic volumes), which therefore is often time consuming. The dynamic programming method only requires an approximate initial horizon that follows the general structure trend, so we can use more efficient interpolation algorithms (e.g., cubic or linear interpolations) to quickly obtain an initial horizon.

We compare the effectiveness of two initialization methods in two examples in Figure 12. The initial horizons are obtained by the slope-based method (the yellow curve) and the cubic interpolation algorithm (the green curve) in Figure 12b and 12e. In the first example (Figures 11b, 11c, and 12a), the refined horizons in Figure 12c are almost identical with different initialization methods. In the second example (Figures 11e, 11f, and 12d) with an obvious fault, Figure 12f also shows comparable results with interpolation (denoted by the green curve) or slope-based initialization (denoted by the yellow curve). Although the interpolation method, compared with the slope-based method, may require more control points (the red circles) to calculate a reasonable initial horizon, it significantly reduces the computational cost and provides convenience for interpreters to implement interactive operations.

Control points

In all examples discussed in the previous sections, the control point, as a crucial constraint, plays an important role in picking a target horizon. Especially in the conventional slope-based method, the horizon exactly passes through those control points, where the control points are required to be accurate. The proposed method also requires some control points in the first initialization step, so we compare the influence of control points on two methods in Figures 13 and 14.

In practice, we expect to select as few points as possible to reduce manual pickings. Therefore, we first investigate the influence of the number of control points for two methods. In Figure 13, we select one, three, and five control points to extract the target horizon with the slope-based method (Figure 13a–13c) and our proposed method (Figure 13d–13f), respectively. Compared to the automatic method based on slopes, our method obtains a more accurate horizon following consistent troughs, particularly in the areas denoted by the blue arrows. We also find that the proposed method (Figure 13d) with only one control point can achieve comparable performance against Figure 13e and 13f. That is, the dynamic programming can refine the horizon to track more consistent phases as well as improve the efficiency by reducing the number of control points.

More importantly, choosing control points in our method can be convenient and efficient because these points only need to be close to the target horizon but are not required to be carefully picked exactly on the horizon. We demonstrate that in Figure 14 in which three control points are picked and two of them (denoted by the purple arrows) are obviously not placed on the target horizon (de-

Methods	Slope calculation	Horizon 1	Horizon 2	Total time
Only slopes	73.1 s	31.07 s	41.92 s	**146.09 s**
Local slopes and multigrid correlations	73.1 s	108.54 s	152.53 s	**334.17 s**
Dynamic programming	------	12.32 s	32.59 s	**44.91 s**

Figure 11. A comparison of computational time for the three different methods.

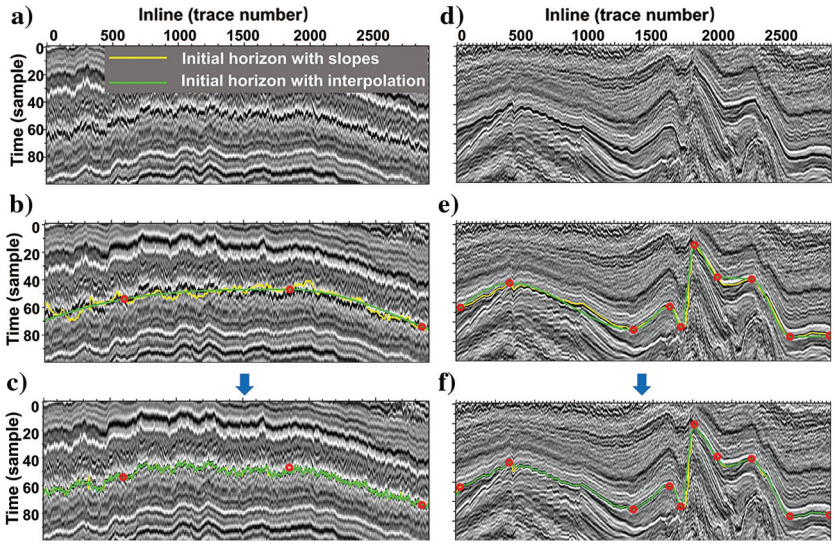

Figure 12. A comparison of two initialization methods. In two input 2D seismic images (a and d), we calculate the initial horizons by the slope-based method (the yellow curves) and the interpolation algorithm (the green curves) in (b and e), respectively. The refined results in (c) show that the proposed method can accurately obtain the optimal path (troughs) by both initialization methods. The results in (f) obtained with interpolation (the green curve) are comparable to those with the slope-based method (the yellow curve). The slope-based method requires estimating reflection slopes and solving for a large nonlinear system, which is often more time-consuming than an interpolation method. On the contrary, the interpolation algorithm is simple and fast even though it may need more control points (the red circles), especially in 3D, to obtain a proper initial horizon.

noted by the blue arrows in Figure 14a). With these control points, the slope-based method obviously fails to pick the target horizon as denoted by the yellow curve in Figure 14a. Our method, however, can still accurately track the target horizon as shown in Figure 14b. In this example, our method effectively corrects the inaccurate initial horizons that are obtained from the inaccurate control points by a simple linear interpolation (the green curve in Figure 14a) or the slope-based method (the yellow curve in Figure 14a). This suggests that the proposed method with both methods of initialization can achieve reasonable results even with incorrect control points because our method does not force the horizon to pass through the selected points.

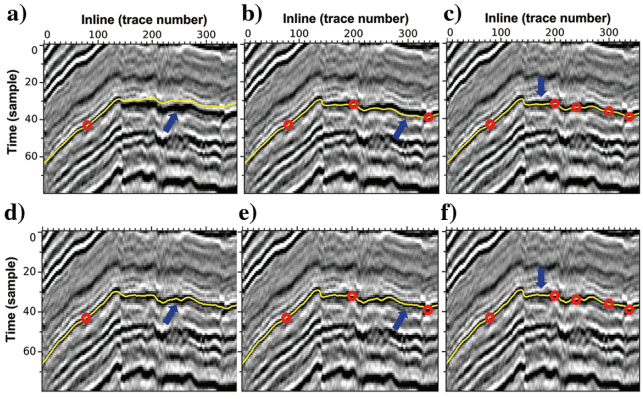

Figure 13. The influence of the number of control points on the initialization step. We select one, three, and five control points (the red circles) to extract the target horizon by (a-c) the slope-based method and (d-f) the proposed method, respectively. Compared to the slope-based method, the dynamic programming obtains a more accurate horizon following the minimum path, especially in the areas denoted by the blue arrows. Also, our method can achieve comparable performance only with one control point in (d). Therefore, our method improves the efficiency by reducing the number of control points.

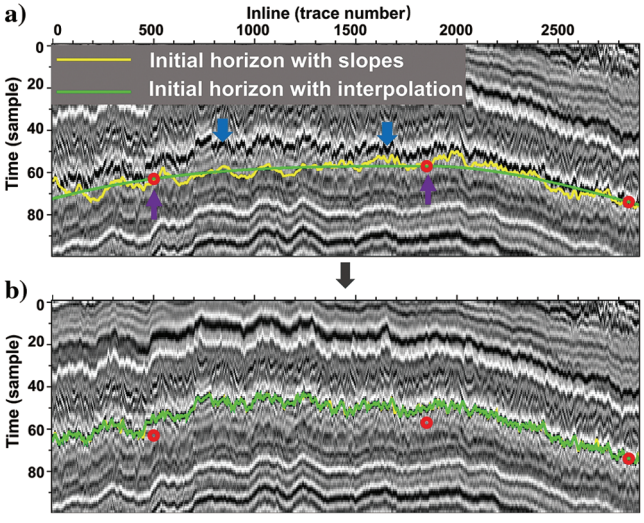

Figure 14. The sensitivity to some incorrect control points. The control points (the red circles) are manually picked and are therefore highly subjective. We give three control points, including two false ones (denoted by the purple arrows). (a) The yellow curve denotes the extracted horizon through the slope-based method, and the green curve denotes the interpolation result with the control points. (b) The refined horizons by our method with initialization by interpolation (the green curve) or the slope-based method (the yellow curve), respectively. This suggests that our method with both methods of initialization can achieve reasonable results even with incorrect control points.

In other words, our method is not sensitive to control points and it only requires a coarse initialization. This aspect greatly improves efficiency and convenience in interpreting seismic horizons. However, if needed, the control points can be set as hard constraints to provide strong control in our horizon refinement method.

Picking window

In the workflow of picking a horizon as described in Figures 5 and 6, we need to choose an appropriate window to extract a subimage. The window is another key factor that will directly affect the results. Because our method refines the horizon by searching the global maximum or minimum values, the window should be chosen to vertically contain the target horizon and to make sure the target horizon is the minimal path within the window. Therefore, as long as the window vertically bounds the target horizon, a vertically smaller window is typically better to achieve higher efficiency and accuracy.

In cases in which the horizons are vertically very close to each other and the nearby horizons show comparable amplitude magnitude to the target horizon, we especially need to carefully choose a vertically narrow window and specify control points as hard constraints (as shown in Figure 6b–6d) to pick the target horizon. In this paper, we used only seismic amplitudes to refine horizons. However, our horizon refinement is not limited to using amplitudes. We can also use other attributes including waveform similarity as the input to our method to obtain the optimal horizons. Using waveform similarity, instead of seismic amplitudes, might be helpful to deal with cases of closely aligned horizons.

In cases in which we deal with large vertical offsets, such as on major faults, we can pick more control points to constrain the target horizon near a fault. Then, we can obtain a much closer initialization to choose a proper window. However, it rarely increases the calculational cost because the interpolation with control points is very fast.

Accuracy and robustness evaluation

We use a 2D example to further evaluate the robustness of our method to noise. We manually pick six horizons as the ground truth as denoted by the yellow curves in Figure 15b. These horizons are dislocated by three faults marked in Figure 15a, which are not complicated but still pose challenges to automatic horizon extraction. We simulate different levels of noise and add them to the original image (Figure 15b). The signal-to-noise ratio (S/N) is 20, 8, 4, 1, and 0.75 in Figure 15c–15g, respectively. In these images, we choose the same control points (the red circles) to pick the target horizons.

With the decreasing the S/N, the proposed method can still obtain reasonable results except for several areas denoted by the red arrows. The horizon picking in these areas can be improved by using more control points as shown in Figure 15h. Compared to the slope-based method, our method obtains more accurate results in areas as denoted by the blue arrows and can follow more consistent phases when passing through faults (the yellow rectangles in Figure 15c).

In Figure 16, we further quantitatively assess the accuracy. The solid and dashed curves in Figure 16a and 16b represent errors with the dynamic programming and the slope-based method, respectively. The error is calculated as follows:

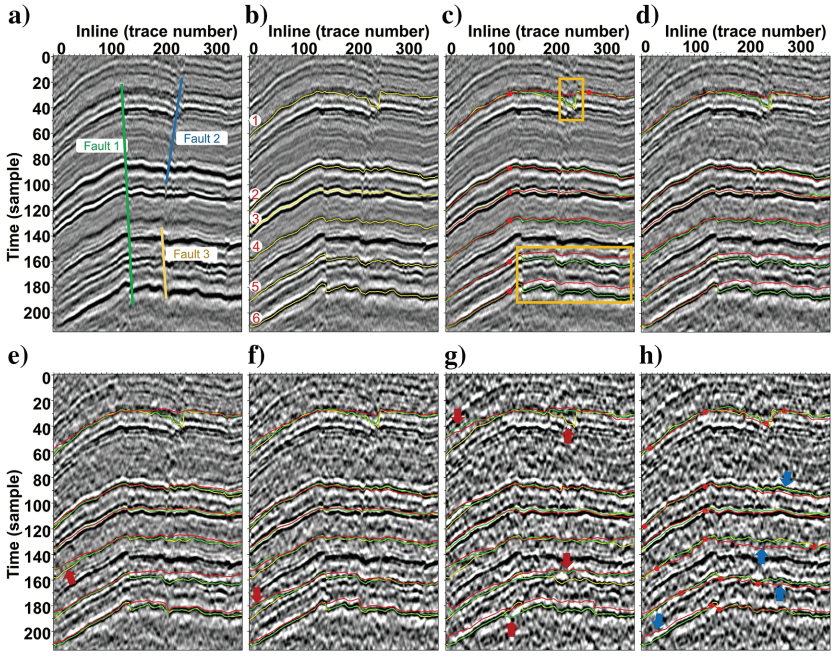

Figure 15. Horizon extraction at different levels of noise. (a) The input image with three faults. (b) Six horizons (the yellow curves in [b]) are manually picked as the ground truth. These target horizons are calculated with the slope-based method (denoted by the red curves) and the dynamic programming (denoted by the green curves), respectively. The S/N in (c-f) is 20, 8, 4, 1, and 0.75, respectively. (h) The image with the same noise as (g), but six horizons are obtained with more control points (the red circles).

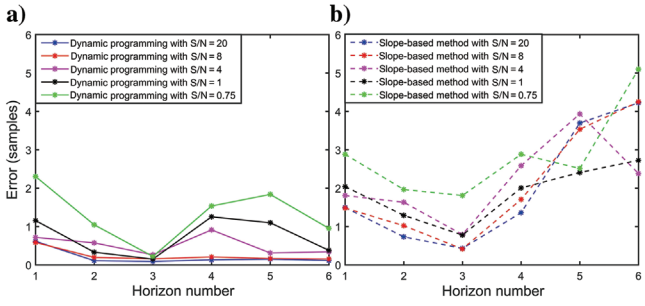

Figure 16. The proposed method generally yields more accurate results (as shown in [a]) than the slope-based method (as shown in [b]) at different levels of noise.

$$E = \frac{\sum_{i=1}^{N} |Z_{\text{true}}[i] - Z_{\text{cal}}[i]|}{N}, \tag{11}$$

where N is the number of samples of each horizon, Z_{true} is the ground truth of a target horizon, and Z_{cal} is a horizon calculated by the slope-based method or our proposed method with dynamic programming. The error of the horizons is mainly affected by faults (e.g., horizons 1, 5, and 6) and noise (e.g., horizon 4). The curves show that the proposed method yields more accurate results passing through faults and is more robust to noise than the slope-based method.

CONCLUSION

We propose a completely new method with the dynamic programming algorithm to search the globally optimal path (two dimensions) or surface (three dimensions) from seismic images, which is highly efficient and can obtain accurate horizons that follow consistent phases even passing through faults.

In our method, we first use manually picking, an automatic method, or interpolation algorithms to compute an initial horizon with some control points. Such an initial horizon requires only an approximate position to follow the trend of the structure. Therefore, we suggest using an interpolation method to obtain an initial horizon, which is typically more efficient than an automatic horizon extraction method (e.g., the slope-based method). We then use the dynamic programming to refine the initial horizon to be more accurate and reveal more detailed geologic structures. During this step, we choose a suitable window centered at the initial horizon and flatten the chosen image according to the initial horizon. In this flattened seismic image, we use the dynamic programming algorithm to efficiently pick the minimal path (two dimensions) or surface (three dimensions). Finally, we map the minimal path or surface from the flattened space back to the original space to obtain the finally updated horizon.

Compared with the widely used slope-based method, our method can follow more consistent phases, reveal detailed structures (e.g., channels), and show effectiveness across faults. Our method also requires only a limited number of control points and is not sensitive to control points, which allows high efficiency and convenience for the horizon interpretation. In some complicated areas, we may impose control points as hard constraints in our method to extract more reliable results.

The proposed method is a completely new method that does not rely on other published methods. The examples demonstrate that it is computationally efficient and robust to noise in the data. Therefore, it is suitable to be implemented as an interactive method to quickly update the horizon while editing the control points, or as a refinement tool for a more hands-on approach. In addition, our method can be used to improve other fully automated workflows (e.g., volumetric slope estimation methods and RGT volumes).

In this paper, we use a seismic amplitude image as an input for horizon extraction by assuming that a horizon follows consistent phases or amplitude values. This assumption, however, is not necessarily true in some complicated cases in which phases may vary along a horizon. In these cases, we may need to use some other proper seismic attributes (instead of amplitude) in which the target horizons can be tracked as globally maximum or minimum paths (two dimensions) or surfaces (three dimensions). Besides, our method extracts a single horizon from a seismic image at one time. We will consider improving our method to simultaneously pick multiple horizons.

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

Arias, E., 2016, Estimating seismic reflection slopes: Master's thesis, Colorado School of Mines.

Bakker, P., 2002, Image structure analysis for seismic interpretation: Ph.D. thesis, Delft University of Technology.

Bakker, P., L. J. van Vliet, and P. W. Verbeek, 1999, Edge preserving orientation adaptive filtering: IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1, 540.

Bellman, R. E., and S. E. Dreyfus, 1962, Applied dynamic programming: Princeton University Press.

Borgos, H. G., T. Skov, T. Randen, and L. Sonneland, 2003, Automated geometry extraction from 3D seismic data: 73rd Annual International Meeting, SEG, Expanded Abstracts, 1541–1544, doi: 10.1190/1.1817590.

Bugge, A. J., S. R. Clark, J. E. Lie, and J. I. Faleide, 2018, A case study on semiautomatic seismic interpretation of unconformities and faults in the southwestern Barents Sea: Interpretation, **6**, no. 2, SD29–SD40, doi: 10.1190/INT-2017-0152.1.

Bugge, A. J., J. E. Lie, A. K. Evensen, J. I. Faleide, and S. Clark, 2019, Automatic extraction of dislocated horizons from 3D seismic data using nonlocal trace matching: Geophysics, **84**, no. 6, IM77–IM86, doi: 10.1190/geo2019-0029.1.

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein, 2001, Introduction to algorithms, 2nd ed.: MIT Press.

de Groot, P., A. Huck, G. de Bruin, N. Hemstra, and J. Bedford, 2010, The horizon cube: A step change in seismic interpretation!: The Leading Edge, **29**, 1048–1055, doi: 10.1190/1.3485765.

Di, H., D. Gao, and G. AlRegib, 2018, 3D structural-orientation vector guided autotracking for weak seismic reflections: A new tool for shale reservoir visualization and interpretation: Interpretation, **6**, no. 4, SN47–SN56, doi: 10.1190/INT-2018-0053.1.

Dorn, G. A., 2011, A review of "global" interpretation methods for automated 3D horizon picking: The Leading Edge, **30**, 38–47, doi: 10.1190/1.3535431.

Equinor, 2018, Volve data village dataset: released under a license based on CC BY 4.0., https://data.equinor.com/.

Fehmers, G. C., and C. F. W. Höcker, 2003, Fast structural interpretation with structure-oriented filtering: Geophysics, **68**, 1286–1293, doi: 10.1190/1.1598121.

Fomel, S., 2002, Applications of plane-wave destruction filters: Geophysics, **67**, 1946–1960, doi: 10.1190/1.1527095.

Fomel, S., 2010, Predictive painting of 3D seismic volumes: Geophysics, **75**, no. 4, A25–A30, doi: 10.1190/1.3453847.

Hale, D., 2009, Structure-oriented smoothing and semblance: CWP Report 635.

Hale, D., 2013, Dynamic warping of seismic images: Geophysics, **78**, no. 2, S105–S115, doi: 10.1190/geo2012-0327.1.

Labrunye, E., and C. Carn, 2015, Merging chronostratigraphic modeling and global horizon tracking: Interpretation, **3**, no. 2, SN59–SN67, doi: 10.1190/INT-2014-0130.1.

Labrunye, E., C. Winkler, C. Borgese, J. Mallet, and S. Jayr, 2009, New 3D flattened space for seismic interpretation: 79th Annual International Meeting, SEG, Expanded Abstracts, 1132–1136, doi: 10.1190/1.3255052.

Lacaze, F. P. S., and T. Valding, 2009, A global approach in seismic interpretation based on cost function minimization: 79th Annual International Meeting, SEG, Expanded Abstracts, 2592–2596, doi: 10.1190/1.3255384.

Lomask, J., A. Guitton, S. Fomel, J. Claerbout, and A. A. Valenciano, 2006, Flattening without picking: Geophysics, **71**, no. 4, P13–P20, doi: 10.1190/1.2210848.

Lou, Y., B. Zhang, T. Lin, N. Liu, H. Wu, R. Liu, and D. Cao, 2019, Accurate seismic dip and azimuth estimation using semblance dip guided structure tensor analysis: Geophysics, **84**, no. 5, O103–O112, doi: 10.1190/geo2018-0530.1.

Luo, S., and D. Hale, 2013, Unfaulting and unfolding 3D seismic images: Geophysics, **78**, no. 4, O45–O56, doi: 10.1190/geo2012-0350.1.

Marfurt, K. J., 2006, Robust estimates of 3D reflector dip and azimuth: Geophysics, **71**, no. 4, P29–P40, doi: 10.1190/1.2213049.

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.

Monniron, M., S. Frambati, S. Quillón, Y. Berthoumieu, and M. Donias, 2016, Seismic horizon and pseudo-geological time cube extraction based on a riemmanian geodesic search: IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 1–5.

Morelatto, R., and R. Biloti, 2013, Structure enhancing filtering with the structure tensor: 83rd Annual International Meeting, SEG, Expanded Abstracts, 4352–4356, doi: 10.1190/segam2013-0615.1.

Parks, D., 2010, Seismic image flattening as a linear inverse problem: Master's thesis, Colorado School of Mines.

Phillips, M., and S. Fomel, 2016, Seismic time-lapse image registration using amplitude-adjusted plane-wave destruction: 86th Annual International Meeting, SEG, Expanded Abstracts, 5473–5478, doi: 10.1190/segam2016-13773139.1.

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.

Qayyum, F., C. Betzler, and O. Catuneanu, 2017, The Wheeler diagram, flattening theory, and time: Marine and Petroleum Geology, **86**, 1417–1430, doi: 10.1016/j.marpetgeo.2017.07.034.

Qayyum, F., C. Betzler, and O. Catuneanu, 2018, Space-time continuum in seismic stratigraphy: Principles and norms: Interpretation, **6**, no. 1, T97–T108, doi: 10.1190/INT-2017-0061.1.

Qayyum, F., O. Catuneanu, and P. de Groot, 2015, Historical developments in Wheeler diagrams and future directions: Basin Research, **27**, 336–350, doi: 10.1111/bre.12077.

Qayyum, F., P. de Groot, and N. Hemstra, 2012, Using 3D Wheeler diagrams in seismic interpretation — The HorizonCube method: First Break, **30**, 103–109.

Shafiq, M., and G. AlRegib, 2016, Interpreter-assisted tracking of subsurface structures within migrated seismic volumes using active contour: 78th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: 10.3997/2214-4609.201600882.

Shafiq, M., Z. Wang, A. Amin, T. Hegazy, M. Deriche, and G. AlRegib, 2015, Detection of salt-dome boundary surfaces in migrated seismic volumes using gradient of textures: 85th Annual International Meeting, SEG, Expanded Abstracts, 1811–1815, doi: 10.1190/segam2015-5927230.1.

Stark, T. J., 2005, Generation of a 3D seismic "wheeler diagram" from a high resolution age volume: 75th Annual International Meeting, SEG, Expanded Abstracts, 782–785, doi: 10.1190/1.2148275.

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: M 26: Seismic stratigraphy — Applications to hydrocarbon exploration: AAPG Memoirs, 99–116.

Wang, K., K. Wei, K. Deal, and D. Wilkinson, 2015a, 3D seismic horizon extraction with horizon patch constraints: 85th Annual International Meeting, SEG, Expanded Abstracts, 1754–1758, doi: 10.1190/segam2015-5877422.1.

Wang, K., K. Wei, K. Deal, and D. Wilkinson, 2015b, A new method of generating horizons from seismic images with geologic constraints: 77th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: 10.3997/2214-4609.201412612.

Wang, Z., and G. AlRegib, 2014, Automatic fault surface detection using 3D Hough transform: 84th Annual International Meeting, SEG, Expanded Abstracts, 1439–1444, doi: 10.1190/segam2014-1590.1.

Wang, Z., Z. Long, and G. AlRegib, 2015c, Tensor-based subspace learning for tracking salt-dome boundaries: IEEE International Conference on Image Processing (ICIP), 1663–1667.

Wang, Z., Z. Long, G. AlRegib, A. Asjad, and M. A. Deriche, 2014, Automatic fault tracking across seismic volumes via tracking vectors: IEEE International Conference on Image Processing (ICIP).

Wu, X., and S. Fomel, 2018, Least-squares horizons with local slopes and multigrid correlations: Geophysics, **83**, no. 4, IM29–IM40, doi: 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.

Wu, X., and D. Hale, 2015, Horizon volumes with interpreted constraints: Geophysics, **80**, no. 2, IM21–IM33, doi: 10.1190/geo2014-0212.1.

Wu, X., S. Luo, and D. Hale, 2016, Moving faults while unfaulting 3D seismic images: Geophysics, **81**, no. 2, IM25–IM33, doi: 10.1190/geo2015-0381.1.

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.

Yu, Y., C. Kelley, and I. Mardanova, 2013, Volumetric seismic dip and azimuth estimation with 2D log-Gabor filter array: 83rd Annual International Meeting, SEG, Expanded Abstracts, 1357–1362, doi: 10.1190/segam2013-0046.1.

Zinck, G., M. Donias, J. Daniel, S. Guillon, 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.

Biographies and photographs of the authors are not available.