Noise-robust detection and tracking of salt domes in postmigrated volumes using texture, tensors, and subspace learning

Zhen Wang¹, Tamir Hegazy¹, Zhiling Long¹, and Ghassan AlRegib¹

ABSTRACT

The identification of salt-dome boundaries in migrated seismic data volumes is important for locating petroleum reservoirs. The presence of noise in the data makes computer-aided salt-dome interpretation even more challenging. We have developed noise-robust algorithms that could label boundaries of salt domes effectively and efficiently. Our research is twofold. First, we used a texture-based gradient to accomplish salt-dome detection. We found that by using a dissimilarity measure based on the 2D discrete Fourier transform, the algorithm was capable of efficiently detecting salt-dome boundaries with accuracy. At the same time, our analysis determined that the proposed algorithm was robust to noise. Once the detection is performed for an initial 2D seismic section, we track the initial boundaries through the data volume to accomplish an efficient labeling process by avoiding the parameter tuning that would have been necessary if detection had been performed for every seismic section. The tracking process involves a tensor-based subspace learning process, in which we built texture tensors using patches from different seismic sections. To accommodate noise components with various levels in a texture tensor, we used noise-adjusted principal component analysis, so that principal components corresponding to greater signal-to-noise-ratio values might be selected for tracking. We validated our detection and tracking algorithms through experiments using seismic data sets acquired from the Netherlands offshore F3 block in the North Sea with very encouraging results.

INTRODUCTION

The deposition of salt in marine basins commonly intrudes into surrounding rock strata and forms an important geologic structure, salt domes. Because of the impermeability of salt, salt domes may seal porous reservoir rocks and lead to the formation of petroleum reservoirs. To localize the possible positions of reservoir regions, experienced interpreters need to accurately label the boundaries of salt domes in migrated seismic data. With the dramatically increasing amount of acquired seismic data, however, manual interpretation is becoming time consuming and labor intensive.

To speed up interpretation, in recent years, researchers have proposed various interactive methods that assist with the detection of salt-dome boundaries using graph theory and image processing techniques. Lomask et al. (2004) define seismic sections as weighted undirected graphs and apply normalized cut image segmentation (NCIS) (Shi and Malik, 2000) to globally optimize the delineation of salt-dome boundaries. As an extension of this method, Lomask et al. (2007) use local dips in the weight matrix and bound constraints to remove boundary artifacts. Similarly, Halpert et al. (2009) introduce the modified NCIS by combining multiple seismic attributes with adaptive weights. Although these NCIS-based methods can be implemented in parallel, the high computational cost limits their future application on high-resolution or 3D seismic data. To improve the efficiency of global segmentation, Halpert et al. (2010) propose to detect the boundaries of salt domes using a pairwise region comparison based on the minimum spanning tree (Felzenszwalb and Huttenlocher, 2004), which reduces the algorithm complexity from $O(n^2)$ to $O(n \log n)$.

In addition to segmentation-based methods, edge detection has also been explored. Zhou et al. (2007) and Aqrawi et al. (2011) apply dip-guided 2D and 3D Sobel filters to detect salt-dome boundaries in time slices. In addition, Berthelot et al. (2013) propose to delineate salt domes by testing a combination of multiple attributes (i.e., texture, frequency, and dip) in a supervised Bayesian classification model. To characterize the homogeneous texture of salt...
bodies, most recently, Hegazy and AlRegib (2014b) derive a directionality attribute from the moment of inertia tensor defined by gradient components. Although the methods mentioned above may be able to achieve various levels of improvement in their detection performance, normally they involve many parameters to be manually tuned, which unavoidably compromises the overall efficiency. In addition to the effectiveness and efficiency issues, salt-dome interpretation may be further complicated by the presence of noise. Typically, noise from different sources is present, which may vary across a seismic volume, thus being space-time dependent. Accommodating such a noise effect is a challenging task for computer-aided interpretation.

In our work, we view seismic volumes as big data volumes containing certain information that needs to be extracted manually or automatically. Hence, we import our long experience in digital signal processing and visual data analysis to seismic interpretation. In our view, a large seismic volume contains information about salt-dome structures. Such a structure is characterized by certain attributes. However, calculating these attributes accurately and efficiently faces many challenges. First, the data are subjected to a long processing pipeline of stacking and migration. Not only do these steps introduce artifacts, but they also may magnify noise in the raw data. Second, the raw data are an undersampled depiction of the subsurface structure. This is analogous to detecting detailed activities from low-quality videos captured with a low-resolution camera. Third, noise in the data volume can be a function of the subsurface structure, and it may contain information about the subsurface structure. Instead of denoising, our long-term plan is to take advantage of such noise.

In this paper, we focus on developing noise-robust algorithms that can label boundaries of salt domes effectively and efficiently. Our research is twofold. First, we define a texture-based attribute, namely, the gradient of texture (GoT), and combine it with a dissimilarity measure based on the 2D discrete Fourier transform (DFT) to implement an algorithm capable of detecting salt-dome boundaries with high accuracy and efficiency. We further analyze and show that the proposed algorithm is robust to noise. Once salt-dome detection is performed for an initial 2D seismic section, we propose to track the initial boundaries through the data volume to accomplish the labeling process. This way, we can avoid tweaking parameters for boundary detection in every seismic section, thus being more efficient. The tracking involves a tensor-based subspace learning process. It starts by forming texture tensors according to the initially labeled boundaries, to better capture the strong correlation between patches of similar texture patterns. It then projects the initial boundaries onto the neighboring seismic section, and it determines the tracked boundaries by evaluating texture similarity in the subspace of the texture tensors. To maintain consistency, we rename the initial and neighboring seismic sections as the reference and predicted sections, respectively, in the rest of this paper. To apply tensor-based subspace learning, we also take noise into consideration. Unlike traditional applications, with seismic data, it is likely that noise of different levels may be included in the same texture tensor. To handle such situations, we use noise-adjusted subspace learning techniques, such as those presented by Roger (1994) and Chang and Du (1999).

An overall block diagram for our salt-dome boundary labeling system is depicted in Figure 1. It includes the detection of salt-dome boundaries for the reference sections and the tracking of the boundaries through the predicted sections. This figure will be used throughout the paper. The rest of the paper is organized as follows: First, we discuss our salt-dome detection method. Then, we present our salt-dome tracking algorithm. Thereafter, we conduct experiments to evaluate the performance of the proposed methods and the noise effect. Based on the results, we draw conclusions at the end of the paper.

THEORY

Salt dome detection

The top boundary of a salt dome usually has a cap rock, which appears in migrated seismic surveys as areas with high contrast and strong edges (Figure 2). However, the lateral boundaries usually lack such high contrast and are more difficult to detect. Figure 2 shows that the lateral boundaries of the salt dome have no explicit edge existing between the salt region and the nonsalt region containing horizons. However, seismic interpretation experts would still be able to visually delineate such boundaries using the change of texture between salt and nonsalt regions shown by other attributes. As a matter of fact, even nonexperts can still perceptually differentiate between the two regions and possibly draw a boundary line. Therefore, we believe that an automated method should take human visual perception into account. Using traditional edge-detection methods (e.g., the Sobel filter) to delineate such a boundary would fail because the boundary exhibits a change in texture, rather than an explicit edge.

![Figure 1](https://library.seg.org/)  
Figure 1. An overall block diagram of the proposed salt-dome interpretation method.
Figure 1 shows an overview of the proposed detection method. To capture the change in texture, we propose a new attribute, which we call GoT. Then, a threshold is applied to the GoT map $G$ to obtain a binary image $B_t$. From an initialization point inside the salt dome, we perform region growing such that the grown region boundaries stop at the threshold value of GoT. We perform binary morphological operations on the grown region to enhance it. The boundary of the salt region is determined based on the enhanced growth region. In the following paragraphs, we explain each of the modules in detail.

In relation to existing methods from the image-processing literature, we show in the following subsections that our proposed GoT is faster and more general than the “texture gradient” introduced by Martin et al. (2004). First, Martin et al. (2004) compute features along 13 different directions, whereas we show that only two directions (horizontal and vertical) are sufficient. Second, Martin et al. (2004) compute the gradient as the norm of the difference between 13D feature vectors, whereas we directly assess the dissimilarity between the neighborhood windows without computing feature vectors. Finally, Martin et al. (2004) compare texture in two half disks centered around a center point, whereas we use square neighborhood windows sharing a side centered around the center point. Using square (or even rectangular) windows makes it easier to take advantage of existing efficient algorithms, such as the fast Fourier transform to assess dissimilarity. It is noteworthy that comparing texture windows is also used in several other applications. For example, Akl et al. (2014) develop a method for a texture synthesis application, in which it is required to measure the dissimilarity between synthesized texture patches and the input texture to be imitated. First, features such as coherency and orientation are computed. Then, the comparison performed by Akl et al. (2014) measures the dissimilarity using various distance measures. In contrast, our method compares the texture windows directly using a very efficient dissimilarity measure without having to compute feature vectors, which can be very computationally demanding. Further, our method takes into account perceptual similarity not just objective similarity because we believe this is one of the important factors in seismic interpretation.

Gradient of texture

We define the GoT at a given point as the perceptual dissimilarity between two square neighborhood windows that share a side centered around the given point. The shared side is perpendicular to the direction of interest (Gonzalez and Woods, 2006; Petrou and Petrou, 2010). Figure 3 illustrates an example with two texture regions separated by a purely vertical boundary. For a given row, if we slide the center point (and hence the two neighborhood windows in the figure) along the horizontal direction, we expect the GoT profile to follow the curve shown at the bottom of the figure. For other rows, we expect similar curves because the boundary is purely vertical in this example. Theoretically, the highest dissimilarity, and hence the highest GoT value, is obtained when the center point falls exactly on the texture boundary because the left neighborhood window $W_{x-}$ will have completely different texture content from that of the right neighborhood window $W_{x+}$. When we move the center point away from the texture boundary to the position shown in Figure 3, the content of $W_{x+}$ is purely from the right texture region, whereas $W_{x-}$ contains both textures. The partially shared...
content will cause the dissimilarity value, and hence the GoT value, to drop. If we keep moving the center point along with the neighborhood windows away from the boundary, we reach a point at which the contents of $W_{x-}$ and $W_{x+}$ are from the same texture region; i.e., the texture contents will be exactly similar. This exact similarity means zero dissimilarity, and hence, zero GoT.

For the texture boundary in Figure 3, computing the GoT in the horizontal direction is sufficient because the texture boundary is purely vertical. For the general case, we compute the GoT as a vector with components in the horizontal and vertical directions. Figure 4 shows how the GoT is computed for the general case. The solid neighborhood windows are used to compute the horizontal component, whereas the dashed neighborhood windows are used to compute the horizontal component. Mathematically, the GoT vector with components in the horizontal and vertical directions is expressed as follows:

$$G_x[i, j] = d(W_{x-}^{ij}, W_{x+}^{ij}), \quad (1)$$

$$G_y[i, j] = d(W_{y-}^{ij}, W_{y+}^{ij}), \quad (2)$$

$$G[i, j] = (G_x^2[i, j] + G_y^2[i, j])^{1/2}, \quad (3)$$

where $G_x[i, j]$ and $G_y[i, j]$ denote the GoT values at point $[i, j]$ along the horizontal and vertical directions, respectively, $G[i, j]$ represents the GoT value combining the horizontal and vertical components, function $d(\cdot)$ defines a dissimilarity measure, and the four $W^{ij}$ expressions with four different subscripts refer to the four neighborhood windows shown in Figure 4.

With the large-scale variations in texture content, the neighborhood window size used to compute the GoT attribute needs to be carefully chosen. To improve robustness, we use different window sizes (five in our case), and we choose the weighted average of the quantities as the attribute value. The neighborhood window sizes range from $3 \times 3$ to $11 \times 11$. Figure 5 shows neighborhood windows of minimum and maximum sizes around a labeled salt-dome boundary. Note that different window sizes enable capturing texture dissimilarity across different scales. The multiscale version of GoT is mathematically expressed as follows:

$$G_x[i, j] = \sum_{n=1}^{N} w_n \cdot d(W_{n,x-}^{ij}, W_{n,x+}^{ij}), \quad (4)$$

$$G_y[i, j] = \sum_{n=1}^{N} w_n \cdot d(W_{n,y-}^{ij}, W_{n,y+}^{ij}). \quad (5)$$

where in our case, $N = 5$. Also, for a given $n$, the neighborhood window size is $(2n + 1) \times (2n + 1)$, and $W_{n,x-}^{ij}$, $W_{n,x+}^{ij}$, $W_{n,y-}^{ij}$, $W_{n,y+}^{ij}$, respectively, denote the neighborhood windows of size $(2n + 1) \times (2n + 1)$ that are to the left, right, bottom, and top of the point $[i, j]$, where the attribute is being computed. Finally, $w_n$ denotes the weight of the GoT value computed at the window, whose size is $(2n + 1) \times (2n + 1)$. The GoT value increases as we increase the neighborhood window size because of the dissimilarity measure we are using (as we show in the next section). Thus, with equal weights, the overall attribute would be more biased to the larger window sizes. As a result, to partially compensate for this bias, we choose $w_n$ to be inversely proportional to $n$ as $w_n = 1/n$.

**Dissimilarity measure**

To measure the perceptual dissimilarity between neighboring windows, we use the following measure:

$$d(W_-, W_+) = E\{|\mathcal{F}\{|\mathcal{F}\{|W_- - W_+\}|\}|\}, \quad (7)$$

where $\mathcal{F}\{\cdot\}$ denotes the 2D DFT, $E$ is the expectation operator, and $W_-$ and $W_+$ denote the two neighboring windows, whose dissimilarity is to be evaluated. This measure is a close variation of the one introduced in Hegazy and AlRegib (2014a), which was shown to be consistent with human perception and computationally efficient. The dissimilarity measure characterizes the amount of variation, or chaos, in the magnitude spectrum of the absolute difference between the two windows. The outer Fourier transform is considered as the magnitude spectrum of the magnitude spectrum. Because the proposed dissimilarity measure is highly consistent with human perception (Hegazy and AlRegib, 2014a), we believe it partially imitates how the human interpreter delineates the salt boundaries according to the texture variations inside and outside the salt region.
Salt dome detection and tracking

Initialization and region growing

GoT detects several texture boundaries including ones outlining salt and others that are irrelevant to salt; e.g., strong reflectors that are away from the salt dome. In order for the detection method to focus on the salt boundaries only, we identify an initialization point inside the salt dome, and then we use the region growing until a predefined threshold for GoT magnitude is met. The initialization point could be selected automatically or interactively with a human interpreter involved. Even with interactive selection of the initialization point, the time spent by a human interpreter will be considerably reduced because the interpreter would have to quickly click on any arbitrary point within the salt region, as opposed to carefully traversing a long, tortuous salt-dome boundary. Note that the proposed method is not sensitive to the initialization point selection as long as the selected point falls anywhere within the salt region. In other words, the initialization point needs to fall within the salt region, but it is otherwise arbitrary.

For automated initialization, we propose a selection method that is based on the directionality attribute (Hegazy and AlRegib, 2014b). The rationale is that salt regions would have the lowest directionality in the whole seismic section under consideration. To compute the directionality map \( D \), we first compute the moment of inertia tensor \( \mathbf{I} \) of the scattered plot of the horizontal and vertical components of the intensity gradient \( \Delta \) computed for a small neighborhood window \( W^{ij} \) centered around the point \([i,j]\). Note that the gradient of intensity \( \Delta \) used here is different from the GoT \( G \) introduced in the previous subsections. Mathematically, we compute the moment of inertia tensor \( \mathbf{I} \) around a given point \([i,j] \) as follows:

\[
\mathbf{I} = \begin{bmatrix}
I_{xx} & I_{xy} \\
I_{yx} & I_{yy}
\end{bmatrix},
\]

\[
I_{xx} = \sum_{[k,l] \in W^{ij}} (\Delta_x[k,l] - \bar{\Delta}_x)^2,
\]

\[
I_{xy} = I_{yx} = \sum_{[k,l] \in W^{ij}} (\Delta_x[k,l] - \bar{\Delta}_x)(\Delta_y[k,l] - \bar{\Delta}_y),
\]

\[
I_{yy} = \sum_{[k,l] \in W^{ij}} (\Delta_y[k,l] - \bar{\Delta}_y)^2,
\]

where

- \( \mathbf{I} \) is the moment of inertia tensor for the scattered plot of the intensity gradient components, which is obtained for a neighborhood window \( W^{ij} \) centered around point \([i,j] \);
- \( W^{ij} \) is a neighborhood window centered around point \([i,j] \);
- the summation is performed over all the points \([k,l] \) in \( W^{ij} \);
- \( \Delta_x[k,l] \) and \( \Delta_y[k,l] \), respectively, denote the horizontal and vertical components of the intensity gradient at a point \([k,l] \) inside \( W^{ij} \);
- \( \bar{\Delta}_x \) and \( \bar{\Delta}_y \), respectively, denote the mean of the horizontal and vertical gradient components computed over the neighborhood window \( W^{ij} \).

The eigenvalues of the moment of inertia tensor correspond to the axes of minimum and maximum moments of inertia. Thus, we can use the eigenvalues of the moment of inertia tensor to capture the directionality of the texture as we show below. To make the method more robust, we average the directionality maps obtained using multiple window sizes as outlined by Hegazy and AlRegib (2014b). Let \( \Lambda_1[i,j] \) and \( \Lambda_2[i,j] \) denote the two eigenvalues of the moment of the inertia tensor computed around point \([i,j] \) using a neighborhood window, whose size is \((2n+1) \times (2n+1)\) and centered around \([i,j] \). We can mathematically define the multiscale directionality attribute at point \([i,j] \) as

\[
D[i,j] = \sum_{n=1}^{N} \left( \frac{\min(\Lambda_1[i,j], \Lambda_2[i,j])}{\max(\Lambda_1[i,j], \Lambda_2[i,j])} \right),
\]

For nonsalt regions, one eigenvalue will be much larger than the other eigenvalue. Thus, the directionality attribute will be close to one. On the other hand, for salt regions, the eigenvalues will be almost equal, leading to directionality attributes close to zero.

For added reliability and to protect against selecting a noisy point that happened to have very low directionality, we smooth out the directionality map with a Gaussian filter. We obtain the initialization point by minimizing the smoothed multiscale directionality map. Therefore, the initialization point is selected according to

\[
\begin{aligned}
\left\{ \begin{array}{l}
\text{arg min}_{[i,j]} D * \Omega,

\Omega[u,v] = e^{-\frac{(u-a)^2 + (v-b)^2}{2c^2}}, \quad u,v \in \{1,2,\ldots,2N+1\},
\end{array} \right.
\end{aligned}
\]

where \( \Omega \) is a Gaussian kernel with a size of \((2N+1) \times (2N+1)\) used to smooth out the multiscale directionality attribute map and \( * \) is the 2D convolution operator. The elements in \( \Omega \) indexed by \( u \) and \( v \) are determined by function \( \sigma_d \), which calculates the distance from \([u,v] \) to the center of the Gaussian kernel. In addition, \( \sigma_d \) denotes the standard deviation of the Gaussian kernel.

We grow the initialization point into a region. Because the initialization point is inside the salt dome; i.e., inside a texture region, we expect the texture gradient around the initialization point to be very low (theoretically zero). Thus, the initialization point can be grown because GoT in the neighboring region is less than a certain threshold value (determined according to the next subsection). As we grow the region, we start meeting points that are close to the boundary of the dome, where the GoT value is high. We stop growing along those points that reached the threshold of the GoT map. In the following subsection, we discuss how the threshold value is determined.

Thresholding

The threshold value can be determined interactively with a human interpreter or automatically. For automated threshold value selection, we propose using Otsu’s (1979) method. The algorithm assumes that there are two classes of pixels following the bimodal histogram. The optimal threshold is therefore calculated as the one
that minimizes the intraclass variance, which is the weighted sum of the variances of the two classes. Thus, the optimal threshold is given by

$$
\arg \min_T \left\{ \sigma^2_1(T) \sum_{i=0}^{T-1} p(i) + \sigma^2_2(T) \sum_{i=0}^{N-1} p(i) \right\},
$$

(14)

where $T$ is the threshold value; $p(i)$ is the probability of the intensity value $i$; and $\sigma^2_1(T)$, $\sigma^2_2(T)$ are the variances of the first and second classes, respectively. The first and second classes are determined given the threshold $T$. The optimal threshold value is found by exhaustively searching for $T$ between 0 and $N-1$.

For more efficient implementation, Otsu shows that minimizing the intraclass variance is equivalent to maximizing the interclass variance, which can be simplified as follows:

$$
\arg \max_T \left\{ \left( \sum_{i=0}^{T-1} p(i) \right) \left( \sum_{i=0}^{N-1} p(i) \right) (\mu_1(T) - \mu_2(T)) \right\},
$$

(15)

where $\mu_1(T)$ and $\mu_2(T)$ are the mean values of the first and second classes, respectively, given a threshold value $T$ (Otsu, 1979).

### Noise impact

In this subsection, we discuss the impact of acquisition noise on the migrated seismic data. Behera and Tsvankin (2008) study the impact of shot gathers contaminated with Gaussian noise on the migrated seismic data. Under Gaussian noise, the reflectors in the migrated image are correctly positioned but are affected by noise (Behera and Tsvankin, 2008). In the following analysis, we assume that the random acquisition noise will result in a migrated image that is contaminated with additive random noise.

Let $\hat{W}_1$ and $\hat{W}_2$ be two noise-contaminated adjacent neighborhood windows, whose dissimilarity is to be evaluated to compute the GoT attribute at their center point, such that

$$
\hat{W}_1 = W_1 + n_1,
$$

(16)

$$
\hat{W}_2 = W_2 + n_2,
$$

(17)

where $W_1$ and $W_2$ denote the noise-free windows, and $n_1$ and $n_2$ denote random noise added to $W_1$ and $W_2$, respectively. We rewrite the dissimilarity measure in equation 7 between the noise contaminated windows as follows:

$$
d(\hat{W}_1, \hat{W}_2) = E[|F\{\Delta_{w} + \Delta_{n}\}|].
$$

(18)

For simplicity, we remove the absolute value of the input to the inner 2D DFT because we noticed through experimentation that it does not make any significant difference in the performance of the proposed method.

Using the linearity property, we can express the inner 2D DFT as follows:

$$
F\{\Delta_{w} + \Delta_{n}\} = F\{\Delta_{w}\} + F\{\Delta_{n}\}. 
$$

(19)

Thus, the 2D DFT coefficients are the same as those of the noise-free case $F\{\Delta_{w}\}$, except they are contaminated with the noise signal $F\{\Delta_{n}\}$. According to Schoukens and Renneboog (1986), we can express this noise signal as

$$
F\{\Delta_{n}\} = \sum_{k=0}^{2n} \sum_{l=0}^{2n} a[k, l] + j \sum_{k=0}^{2n} \sum_{l=0}^{2n} b[k, l],
$$

(20)

where each summation is performed over the square neighborhood window size of $(2n+1) \times (2n+1)$ and $a$ and $b$ are random variables. The noise signals affecting the real and imaginary parts of the Fourier coefficients are derived from the sum of several random variables. Therefore, the original Fourier coefficients are subjected to additive random noise. According to Schoukens and Renneboog (1986), the covariance matrix of the real and imaginary parts is nearly diagonal, which means that the real and imaginary coefficient noise components are independent.

The $F\{\Delta_{w}\}$ and $F\{\Delta_{n}\}$ are complex. As a result, the absolute value of $F\{\Delta_{w}\} + F\{\Delta_{n}\}$ can be determined by vector algebra as shown in Figure 6, where $\hat{s}$ refers to $F\{\Delta_{w}\}$ and $\hat{n}$ refers to $F\{\Delta_{n}\}$. If the signal-to-noise ratio (S/N) is much larger than one, then angle $\theta$ in Figure 6 is very small. As a result, noise impact on the absolute value of Fourier coefficients can be approximated by projecting the noise component $\hat{n}$ on the original signal $\hat{s}$ vector direction. Consequently, such noise can be approximated as additive random noise $n_m$ as follows:

$$
n_m = |F\{\Delta_{n}\}| \cos \phi,
$$

(21)

where $\phi$, which is a random variable, denotes the angle between the signal and noise vectors in Figure 6.

Consequently, $|F\{\Delta_{w}\} + F\{\Delta_{n}\}|$, which is the input to the outer 2D DFT, suffers additive random noise $n_m$. Similarly, applying the same derivation for the magnitude of the outer 2D DFT, we conclude that noise on the magnitude of the outer 2D DFT coefficients is additive random noise as well, say $n_d$. Because the dissimilarity measure $d()$ is the mean value of the outer 2D DFT magnitude, then $d()$ is shifted by the mean of $n_d$, that is, $\mu_d$.

Theoretically, the additive random noise affecting the migrated images shifts the dissimilarity measure (and hence, GoT) by a constant value $\mu_d$, assuming the noise is stationary. Because we apply
Salt dome detection and tracking

Salt-dome tracking

In the detection of salt-dome boundaries, the accuracy of labeled boundaries depends greatly on the selection of certain parameters. Because salt-dome structures commonly vary across seismic sections, interpreters have to adjust parameters so that detected boundaries in each section can accurately capture structural changes. However, extra labor and time involved in the tuning of parameters may lower the interpretation efficiency. Therefore, to avoid the frequent adjustment of parameters, we group seismic sections into reference and predicted sections by borrowing the ideas of I- and P-frames from video-coding techniques and apply the proposed detection method only to reference sections that constitute the majority of seismic sections. By tracking detected boundaries in reference sections through a seismic volume, we can synthesize salt-dome boundaries in predicted sections with limited human intervention. The lower part of Figure 1 illustrates the pipeline of the proposed tracking method, each step of which is going to be introduced in detail in the following subsections.

Tensors and multilinear analysis

By observing salt domes in seismic sections as Figure 2 shows, we notice that local areas along salt-dome boundaries commonly have similar textures. Therefore, by classifying boundary textures and use the tensor-based subspace learning algorithm to acquire their texture information, we can build texture tensors and use the tensor subspace learning algorithm to acquire their texture information. In addition, the product above can also be implemented by folding the third-order tensor along three modes. The inverse operation of $\mathbf{U} \cdot \mathbf{A}^{(3)}$ is $\mathbf{B}^{(3)} = \left( \sum_{i_n} \mathbf{A}^{(3)}(i_1 \ldots i_N) \cdot \mathbf{U}(j_n, i_n) \right)$. In addition, the product above can also be implemented by folding the third-order tensor along three modes. The inverse operation of $\mathbf{U} \cdot \mathbf{A}^{(3)}$ is $\mathbf{B}^{(3)} = \left( \sum_{i_n} \mathbf{A}^{(3)}(i_1 \ldots i_N) \cdot \mathbf{U}(j_n, i_n) \right)$.

Texture tensor classification

To extract texture features from salt-dome boundaries detected in a reference section, we define pairs of patches centered at boundary points that are subimages containing the boundary textures of the reference section and its corresponding GoT map. Because matrices are particular third-order tensors with a third-mode dimension equal to one, we denote patch pairs in a form of tensor pairs as $\mathbf{S}_p, \mathbf{G}_p \subset \mathbb{R}^{L \times M \times 1}$, where $N_p$ represents the number of all boundary points and $I_1$ and $I_2$ define the dimensions of patches along the depth and crossline directions, respectively. Because textures have no bias on either the depth or crossline directions, we commonly assume that $I_1$ is equal to $I_2$. In addition, to ensure that boundary textures can be captured by square-shaped patches, we determine the edge length of patches as around one-tenth of the larger of the vertical and horizontal resolutions. Furthermore, by evaluating the similarity between patches, we can classify boundary textures and build a set of texture tensors, denoted $\{\mathbf{S}_k, \mathbf{G}_k\}, k = 1, 2, \ldots, N_t$.

To clarify the iterative classification strategy, we depict its block diagram in Figure 8. Because of the roughly semicircular shape of detected boundaries in the reference section, we traverse clockwise along the boundary points and define $p_1$ as the first point in the bottom-left corner satisfying the dimension constraint of patches. After initializing $\{\mathbf{S}_1, \mathbf{G}_1\}$ with patch pair $\{\mathbf{S}_{p_1}, \mathbf{G}_{p_1}\}$, we search for $p_2$ in the $3 \times 3$ neighborhood of $p_1$ on the basis of the point priority shown in Figure 9a, where a smaller number represents a higher priority. To facilitate the evaluation of the texture similarity between the patch pair $\{\mathbf{S}_p, \mathbf{G}_p\}$, and the initial tensor pair, $\{\mathbf{S}_1, \mathbf{G}_1\}$, we append $\mathbf{S}_{p_1}$ and $\mathbf{G}_{p_1}$ to $\mathbf{S}_1$ and $\mathbf{G}_1$, respectively, along the third mode and obtain the updated tensor pair $\{\mathbf{S}_1, \mathbf{G}_1\}$. In Figure 7, function $\langle \cdot | \cdot \rangle$ describes an operation that extends tensors along the third direction. Furthermore, by applying the MPCA to}

\[ \mathbf{C} = \mathbf{A} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \cdots \times_N \mathbf{U}^{(N)^T}, \]
\[ \{ \hat{S}_1, \hat{G}_1 \} \text{, we obtain the projection matrix of each mode, denoted } \hat{U}^{(n)}_{M_i} \in \mathbb{R}^{N_i \times P_i}, (n = 1, 2) , \ \hat{U}^{(3)}_{M_i} \in \mathbb{R}^{(1 \times M_i \times P_i), M = \{ S, G \}}, \text{ in which } [P_1, P_2, P_3] \text{ are determined empirically. Because of the high computational efficiency of the sequential Karhunen-Loève algorithm (Levey and Lindenbaum, 2000; Hu et al., 2011) on updating projection matrices, we extract } \hat{U}^{(3)}_{M_i}, M = \{ S, G \} \text{ from the row spaces of third-mode unfolding matrices instead of their column spaces. Because eigenvectors in the projection matrices reflect the texture features of tensors, to evaluate the similarity between patches and tensors, we define reconstruction error } e_c \text{ as follows:}
\]
\[
e_c = \sum_{n=1,2, M = \{ S, G \}} ||M_{p_k} - M_{p_k} \times_n (\hat{U}^{(n)}_{M_i} \cdot \hat{U}^{(n)}_{M_i})||^2_F + \sum_{M = \{ S, G \}} ||M_{p_k} - M_{p_k} \cdot \hat{U}^{(3)}_{M_i} \cdot \hat{U}^{(3)}_{M_i}||^2_F, \tag{24}
\]

where \( p_k \) and \( k \) represent the indices of the current boundary point and tensor pair, respectively. If \( e_c \) is less than threshold \( T_c \), then the strong similarity allows us to extend \( \{ S_1, G_1 \} \) by appending \( \{ S_{p_k}, G_{p_k} \} \). Otherwise, the great deviation requires us to keep \( \{ S_1, G_1 \} \) unchanged and initialize another tensor pair \( \{ S_2, G_2 \} \) with \( \{ S_{p_k}, G_{p_k} \} \). By repeating the steps above, we obtain classified tensor pairs that contain the local texture information of the detected boundary. Figure 9b illustrates one local area of a seismic section and its corresponding GoT map, in which the white curve represents the labeled salt-dome boundary. In addition, Figure 9b also shows the third-mode unfolding matrices of a tensor pair extracted from this local area, in which texture patches have a strong correlation.

**Tracked boundary synthesis**

On the basis of texture features extracted from classified texture tensors, we can localize boundary points in predicted sections. We first project the detected boundary in the reference section onto the target section and keep the coordinates of all points unchanged. Then, to identify the tracked point of every projected point \( p \), we search among candidate points located along the normal direction of the projected boundary within radius \( R_s \), which is adaptively determined by the inline number difference between the reference and the current predicted sections. To define the textures of candidate points, we extract pairs of patches centered at these points from the predicted section and its corresponding GoT map, denoted as \( \{ S_{p,j}, G_{p,j} \} \in \mathbb{R}^{(1 \times M_i \times N_i)}, j = 1, 2, \ldots, (2R_s + 1) \), where \( j \) represents the index of the candidate point. Because in the reference section of the patch pair of projected point \( p \) belongs to tensor pair \( \{ S, G \} \), by comparing the similarity between \( \{ S_{p,j}, G_{p,j} \} \) and \( \{ S, G \} \), we can determine tracked point \( p^* \). The block diagram of the proposed tracking process is shown in Figure 10. The reconstruction error \( e_r \) is calculated as follows:

\[
e_r = \sum_{n=1,2, M = \{ S, G \}} \lambda_M \cdot ||M_{p,j} - M_{p,j} \times_n (\hat{U}^{(n)}_{M_i} \cdot \hat{U}^{(n)}_{M_i})||^2_F + \sum_{M = \{ S, G \}} \lambda_M \cdot ||M^{(3)}_{p,j} - M^{(3)}_{p,j} \cdot \hat{U}^{(3)}_{M_i} \cdot \hat{U}^{(3)}_{M_i}||^2_F, \tag{25}
\]

where \( \lambda_S = 1 \) and \( \lambda_G = ||\log \{ G[i,j] \}|| \) represent the weights of texture features extracted from seismic sections and corresponding GoT maps, respectively, and \( [i,j] \) denote the coordinates of the

**Figure 8.** The block diagram of the adaptive classification of texture tensors.

**Figure 9.** (a) The priority of points in the 3 \times 3 neighborhood of \( p_1 \) and (b) the third-mode unfolding matrices of a tensor pair extracted from the local area of a seismic section and its corresponding GoT map.
Salt dome detection and tracking

Candidate point. Because Got maps have been normalized to a range of zero to one, a larger $G[i,j]$ indicates a smaller weight, which can move tracked points toward the real salt-dome boundaries of predicted sections by lowering the reconstruction error. Finally, the candidate point with the smallest reconstruction error is selected as the tracked point. We apply the same process to all boundary points and obtain binary image $B$, which contains the initial tracked points.

Because of the complicated structures of salt domes, it is inevitable that tracked points involve some outliers that may harm the smoothness and continuity of tracked boundaries. To remove these outliers, we apply the median filter to $B$ and obtain filtered image $\tilde{B}$ as follows:

$$
\tilde{B}[i,j] = \begin{cases} 
0 & \text{if } (i,j) \text{ satisfies the constraint in equation 28, otherwise,} \\
\text{median}(B[i,j]) & \text{if } (i,j) \text{ does not satisfy the constraint in equation 28,}
\end{cases}
$$

where function $\text{median}(\cdot)$ applies the median filter to the $R_m \times R_m$ analysis window of point $(i,j)$. Without involving new points generated by median filtering, equation 26 removes outliers and ensures the accuracy of the remaining tracked points. Furthermore, we connect the remaining tracked points clockwise with straight lines and obtain the initial labeling of the salt-dome boundary. Because the connection based on lines may result in a jagged shape of the labeled boundary, which is geologically unreasonable, to delineate the salt-dome boundary more accurately, we need to use two postprocessing steps: dilation and skeletonization. We dilate initially labeled result $L$ using disk-shaped structural element $H$ as follows:

$$
L \oplus H = \bigcup_{z \in L} H_z, \quad (27)
$$

where $H_z$ represents the structural element centered at 2D point $z$. The dilated region can be understood as the locus of the points covered by $H$, when the center of $H$ moves inside $L$. Furthermore, by applying skeletonization (Kong and Rosenfeld, 1996) to the dilated region, we can extract the tracked boundary.

Salt-dome tracking based on noise-adjusted MPCA

In our previous discussion, we commonly assume that noise does not exist in seismic data sets. However, in real cases, it is possible that migrated seismic sections contain various levels of noise. If we still depend on the MPCA-based tracking method, extracted features corresponding to the most variations of texture tensors may have low S/N. Therefore, the appearance of noise may reduce the accuracy of tracked boundaries. To deal with this problem, on the basis of Rogers’s (1994) work, we modify the proposed tracking strategy and introduce noise-adjusted MPCA (NA-MPCA), which ranks principal components (PCs) along different modes by the S/N rather than the variation. Because noise is commonly assumed to be additive, we define image patches extracted from seismic sections as $X = S + N$, where $S$ is the image patch without noise and $N$ represents noise that are independent to $S$. In this paper, we estimate the amplitude of noise using a wavelet-based denoising method (Chang et al., 2000). In addition, we denote the covariance matrices of $X$, $S$, and $N$ as $\Sigma_X$, $\Sigma_S$, and $\Sigma_N$, respectively. To arrange PCs in the order of S/Ns, we need to apply a noise-whitening process that transforms $N$ to identify covariance matrix $\hat{N}$. Therefore, we define transform matrix $P$ and analyze the noise-whitening process as follows:

$$
\Sigma_N = P^T \Sigma_N P = P^T \Delta_N V^T P = I, \quad (28)
$$

where orthonormal matrix $V$ contains the eigenvectors of $\Sigma_N$. To satisfy the constraint in equation 28, we yield transform matrix $P = V \Delta_N^{1/2}$. Then, we implement the noise-whitening process in the proposed salt-dome tracking method by replacing the image patch $S_p, i = 1, 2, \ldots, N_b$, with $S_p = X_i P$ and keeping $G_p$, unchanged because of the blurring effect of noise to the Got maps. NA-MPCA removes the influence of different noise levels in the reference and predicted sections and ensures the robustness of the proposed tracking algorithm.

Similarity index measurement of salt-dome boundaries

By using the proposed salt-dome detection and tracking method, we can semiautomatically delineate salt-dome boundaries in seismic sections. To evaluate the performance of the proposed method, we introduce a salt-dome similarity (SalSIM) index that measures the distances between semiautomatically labeled results and the ground truth provided by experienced interpreters. To explain the definition of the SalSIM index, in Figure 11, we draw two types

![Figure 10. Block diagram of the identification of tracked points.](source-url)
of curves for illustration. The proposed SalSIM index is developed based on the Fréchet distance (Alt and Godau, 1995), which can more accurately measure the deviation between two curves than the Hausdorff distance (Huttenlocher et al., 1993) by taking the continuity of curves into account.

On the basis of the Fréchet distance, we attempt to numerically describe local and global deviations between labeled results and the ground truth using local and global items in the proposed SalSIM index, respectively. To compare the local details of labeled boundaries and the ground truth, we define an analysis window that identifies a pair of curve segments, as the rectangle shown in Figure 11. Therefore, by moving the analysis window along the ground truth and calculating the Fréchet distance of every pair of local segments, we can obtain a sequence of distances \( d = [d_i], i = 1, 2, \ldots, N_d \), where \( N_d \) is the total number of pairs. The mean and standard deviation of \( d \), denoted \( \mu_d \) and \( \sigma_d \), respectively, constitute the local item of the proposed SalSIM index. In addition, the global item involves the Fréchet distance of two entire curves, denoted \( d_{\text{max}} \), which equals the largest distance in \( d \). Therefore, on the basis of obtained statistics, we introduce the SalSIM index as

\[
\text{SalSIM} = e^{-\alpha \mu_d - \sigma_d} \cdot e^{-\beta d_{\text{max}}},
\]

(29)

where \( \alpha \) and \( \beta \) are normalization factors determined empirically. Because the exponential function defined on negative real numbers has a range of zero to one, we apply it in the SalSIM index for normalization. Therefore, according to the expression of SalSIM in equation 29, a greater SalSIM value represents smaller deviation between the semiautomatically labeled salt-dome boundary and the ground truth and vice versa.

**EXAMPLES**

In this section, we evaluate the performance of our proposed detection and tracking method on a real seismic volume acquired from the Netherlands offshore F3 block with a size of \( 24 \times 16 \) km in the North Sea (OpenDect, 1987). We focus on a local volume containing a distinguishable salt dome that has an inline number ranging from 389 to 409, a crossline number ranging from 401 to 701, and the time interval between 1300 and 1848 ms sampled every 4 ms. Figure 12 illustrates a seismic section, inline 400, extracted from the local volume.

![Figure 11. An example of the salt-dome boundary labeled by a computer-aided interpretation method and the ground truth labeled manually.](Image)

**Salt-dome detection**

The proposed salt-dome detection method delineates the boundaries of salt domes based on the GoT maps of seismic sections, in which the point value represents the dissimilarity of neighboring square windows. To capture the multiscale texture contrast of neighborhoods, we calculate GoT values by varying the window size from \( 3 \times 3 \) to \( 11 \times 11 \) and averaging the corresponding dissimilarity measures, denoted \( G \) in equation 3. Figure 13a shows the GoT map of inline 400, in which likely boundary regions have higher GoT values. On the basis of GoT maps, we use the region-growing method to identify salt-dome areas and their corresponding boundaries. In the growing process, we tweak threshold \( T_g \) for each seismic section to yield the best growing result. In Figure 13b, we compare the green detected boundary with the manually labeled red ground truth and notice that the former overlaps the most part of the latter except the bottom-left and -right corners.

**Subjective comparison of detected boundaries**

Because GoT maps mainly characterize the texture variations of local neighborhoods in seismic sections, to demonstrate the benefit of the GoT attribute, we compare it with two conventional seismic attributes, the GLCM contrast attribute (Haralick et al., 1973) and the gradient attribute (Toriwaki and Yoshida, 2009), both of which have been introduced to detect salt-dome boundaries (Aqrawi et al., 2011; Berthelot et al., 2013). To ensure a fair comparison, we apply the same region-growing method to extract salt-dome boundaries from different attribute maps. The GLCM describes the distribution of co-occurring gray-scale values at a given offset over an image. For each point \([i, j]\), within its \((2R_d + 1) \times (2R_d + 1) \) analysis window, by varying the directions and pixel distances of offset \([\Delta i, \Delta j] \), we can obtain a series of GLCMs, denoted \( P_{[\Delta i, \Delta j]} \). Therefore, on the basis of these GLCMs, we define the GLCM contrast attribute \( C[i, j] \) as

\[
C[i, j] = \sum_{[\Delta i, \Delta j]} \left( \frac{1}{N_g} \sum_u \sum_v (u - v)^2 P_{[\Delta i, \Delta j]}[u, v] \right),
\]

(30)

where \( u \) and \( v \) indicate the row and column indices of the GLCM, respectively, and \( N_g \) represents the number of all possible offsets. In the tested local volume, we obtain the GLCM contrast maps of seismic sections by setting \( R_d = 4 \). Figure 13c illustrates the GLCM contrast attribute of inline 400, in which higher contrast values indicate likely boundary areas. By applying the region-growing method to Figure 13c, we can label the salt-dome boundary as Figure 13d shows. Similar to the GLCM contrast attribute, the gradient attribute

![Cross section of a salt dome.](Image)
attribute estimated based on the 3D Sobel filter can also identify the boundaries of salt domes. To approximate the partial derivatives of different directions, we convolute three $3 \times 3 \times 3$ kernels of the Sobel filter with the local volume. For each point, the magnitude of the gradient attribute is calculated as follows:

$$F = \sqrt{F_x^2 + F_y^2 + F_z^2},$$  

(31)

where $F_x$, $F_y$, and $F_z$ represent the partial derivatives of the crossline, depth, and inline directions, respectively. Figure 13e illustrates the gradient map of inline 400, from which we extract the green salt-dome boundary as Figure 13f depicts.

By comparing the GoT and GLCM contrast of inline 400 in Figure 13, we notice that it is not easy to determine which one leads to the more accurate detection of salt-dome boundaries because of their comparable performance on highlighting boundary areas. However, without multiscale-based description, the gradient map depending only on $3 \times 3$ local regions shows noisy and discontinuous stripes around boundary areas, which limits the accuracy of detected boundaries. Figure 14 illustrates the local regions of Figure 13b, 13d, and 13f, in which every column contains the same local regions of salt-dome boundaries detected from various attribute maps and every row contains the different local regions of a detected salt-dome boundary. The first column illustrates the bottom-left corner of detected boundaries, in which the second one is slightly more similar to the ground truth than the other two local boundaries. However, in the second to fourth columns, boundaries detected by the proposed method show the highest accuracy.

**Objective comparison of detected boundaries**

To quantize subjective perception, we use the SalSIM index to evaluate the similarity between the ground truth and the detected boundary. Figure 15 shows the SalSIM indices of the salt-dome boundaries of inline 389–409 detected by various methods. Boundaries extracted from gradient maps have the lowest accuracy because of noise around boundary areas. In most of the tested seismic sections, boundaries detected by the proposed method have SalSIM indices higher than or equal to those obtained from GLCM contrast maps. Because the computation of GLCMs is time consuming, the proposed method outperforms the GLCM-based detection method in efficiency and accuracy. Table 1 lists several statistical measures of SalSIM indices, in which the averaged maximum distance (AMD) in pixels represents the mean of the $d_{max}$ of all detected boundaries. The SalSIM indices of boundaries detected by the proposed method have the largest mean, the smallest standard deviation, and the shortest AMD, which proves the superiority of the proposed method.

To further verify the robustness of the proposed method, we apply the GoT-based detection method on the 10 sections of the F3 block with a crossline number ranging from 834 to 843, inline num-

Figure 13. (a, c, e) Illustrate the GoT, GLCM contrast, and gradient maps of inline 400, respectively; (b, d, f) compare the manually labeled red ground truth with the green salt-dome boundaries detected from attribute maps, panels (a, c, and e), respectively.

Figure 14. Local salt-dome boundaries extracted from the GoT, GLCM contrast, and gradient maps are labeled in green. The ground truth labeled manually is labeled in red.
ber ranging from 279 to 600, and the time interval between 1300 and 1848 ms sampled every 4 ms. Figure 16 illustrates one section of this local data set, and Table 2 shows the performance of various detection methods, in which the proposed GoT-based method still achieves the highest accuracy.

### Robustness to noise

To verify the robustness of the proposed method, we add Gaussian noise to seismic sections and compare the accuracy of salt-dome boundaries detected from GoT, GLCM contrast, and gradient maps.

![Figure 15. The SalSIM indices of salt-dome boundaries detected from GoT, GLCM contrast, and gradient maps.](image)

**Table 1. The statistical measures of SalSIM indices in Figure 15 obtained from various detection methods.**

<table>
<thead>
<tr>
<th>Detection method</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>AMD (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection method based on GoT maps</td>
<td>0.9348</td>
<td>0.0104</td>
<td>11.64</td>
</tr>
<tr>
<td>Detection method based on GLCM contrast maps Berthelot et al. (2013)</td>
<td>0.9290</td>
<td>0.0104</td>
<td>13.92</td>
</tr>
<tr>
<td>Detection method based on gradient maps Aqrawi et al. (2011)</td>
<td>0.9054</td>
<td>0.0159</td>
<td>20.73</td>
</tr>
</tbody>
</table>

![Figure 16. A seismic section (crossline 834) of the crossline volume contains the cross section of a salt dome.](image)

**Table 2. The statistical measures of the SalSIM indices of boundaries delineated by various detection methods in the crossline data set.**

<table>
<thead>
<tr>
<th>Detection method</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>AMD (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection method based on GoT maps</td>
<td>0.9684</td>
<td>0.00280</td>
<td>4.67</td>
</tr>
<tr>
<td>Detection method based on GLCM contrast maps Berthelot et al. (2013)</td>
<td>0.9645</td>
<td>0.00379</td>
<td>5.68</td>
</tr>
<tr>
<td>Detection method based on gradient maps Aqrawi et al. (2011)</td>
<td>0.9396</td>
<td>0.00860</td>
<td>9.98</td>
</tr>
</tbody>
</table>

**Salt-dome tracking**

In the local seismic volume, we define inline 400 as a reference section, the salt-dome boundary of which has been labeled by experienced interpreters or computer-aided detection methods. We can track the reference boundaries through the seismic volume and synthesize salt-dome boundaries in the neighboring 20 predicted sections ranging from inline 389 to 409. Each point at the reference boundary corresponds to a pair of 31 × 31 image patches extracted from the reference section and its corresponding GoT map. To acquire texture features from the local regions of the reference boundary, we group all these patch pairs into tensor pairs on the basis of the block diagram shown in Figure 8.

The dimensions of texture features in the three modes are [15, 15, 5], and the threshold $T_x$ on the reconstruction error is 2.3. Furthermore, in the predicted sections, the proposed tracking method searches along the normal direction of the projected point and identifies the position of the tracked point by comparing the similarity between the patch pairs of candidate points and tensor pairs built from the reference section. The localization of tracked points is implemented automatically, which improves interpretation efficiency. Finally, we remove noisy points in predicted sections with $2 \times 2$ median filters and...
connect remaining points to label the salt-dome boundary under the shape constraint of salt domes. Figure 18a compares the green tracked salt-dome boundary in inline 391, synthesized based on the manually labeled reference boundary in inline 400, with the red ground truth. We notice that these two curves almost overlap except for several local regions.

**Subjective comparison of tracking methods**

To prove that tracking accuracy can be increased by involving the GoT attribute and the texture features of all modes, we need to compare the proposed method with three other tracking strategies. The first one is to implement the tracking process depending only on texture features extracted from vectorized patches rather than those from third-order tensors. The second one is to synthesize tracked boundaries within the framework of the proposed method, but without involving the GoT attribute. The third strategy is the same as the proposed method except for using GLCM contrast maps instead of GoT maps. We rename these three tracking strategies as the tracking method based on vectorization, the tensor-based tracking method without GoT maps, and the tensor-based tracking method with GLCM contrast maps. Figure 18b–18d compares the red ground truth labeled manually with green salt-dome boundaries labeled by the three tracking strategies mentioned above, and the tracked boundary in Figure 18c shows the greatest deviation from the ground truth. Figure 19 illustrates the local regions of Figure 18, in which every column contains the same local regions of salt-dome boundaries synthesized based on various tracking methods, and every row contains the different local regions of a tracked salt-dome.

![Figure 17](image1.png)

**Figure 17.** The averaged SalSIM indices of salt-dome boundaries detected from noisy seismic sections based on the proposed method and the GLCM-based method. (a) The proposed method without denoising. (b) The proposed method based on denoising operation. (c) The GLCM-based detection method.

![Figure 18](image2.png)

**Figure 18.** (a-d) Show the comparison between the red ground truth and the green tracked salt-dome boundaries of inline 391 synthesized by the proposed method, the tracking method based on vectorization, the tensor-based tracking method without GoT maps, and the tensor-based tracking method with GLCM contrast maps, respectively.
boundary. By comparing the local regions of the tracked boundaries with the ground truth, we notice that the boundary in the first row synthesized by the proposed tracking method is the one most similar to ground truth.

Objective comparison of tracking methods

To objectively verify our conclusion, we synthesize tracked boundaries in seismic sections ranging from inline 389 to 409 using various tracking strategies and plot their SalSIM indices in Figure 20. The horizontal and vertical axes represent the inline number and the SalSIM index, respectively. Although a tracking method based on vectorization can achieve comparable accuracy with the proposed method in inline 401–409, the SalSIM indices of salt-dome boundaries in inline 395–389 synthesized by the former decrease more quickly than those obtained by the latter. In addition, the dashed curves in Figure 20 show the distribution trends of the SalSIM indices. Although the SalSIM indices of four tracking strategies have similar trends that drop with the increasing offsets between predicted sections and the reference section, the proposed tracking method outperforms other tracking strategies particularly in inline 389–399. Table 3 contains several statistical measures of SalSIM indices in Figure 20. The mean of SalSIM indices and the AMD evaluates the accuracy of the tracking method, and the standard deviation determines robustness. The proposed method has the greatest mean, the smallest standard deviation, and the shortest AMD, which proves the superiority of the proposed method.

Combination of proposed detection and tracking methods

In the previous sections, tracked boundaries are synthesized based on the boundary labeled by interpreters in the reference section. To further reduce human intervention in the tracking process, in the local seismic volume, we can synthesize the tracked boundaries of inline 389–409 on the basis of the reference boundary in inline 400 labeled by the proposed detection method. Figure 21 compares the tracked and detected green boundaries with the red ground truth. In Figure 21a and 21b, the tracked boundary of inline 392 is more similar to the ground truth than the detected result. In contrast, in Figure 21c and 21d because the bottom-left corner of the tracked boundary has great deviation to the ground truth, the detected boundary in inline 408 seems to be more accurate. However, the appearance of the large deviation is caused by the deviation in the reference boundary shown in Figure 13b rather than our tracking method. To objectively compare the tracked and detected boundaries, we illustrate the corresponding SalSIM indices in Figure 22. Because we use $\lambda_g$ as an important weight that helps to move tracked points to the boundary area, the tracked boundaries may have higher SalSIM indices than

Table 3. The statistical measures of SalSIM indices in Figure 20 obtained from various tracking strategies.

<table>
<thead>
<tr>
<th>Tracking method</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>AMD (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed tracking method</td>
<td>0.9531</td>
<td>0.0111</td>
<td>8.32</td>
</tr>
<tr>
<td>Tracking method based on vectorization</td>
<td>0.9496</td>
<td>0.0148</td>
<td>9.30</td>
</tr>
<tr>
<td>Tensor-based tracking method without GoT maps</td>
<td>0.9364</td>
<td>0.0295</td>
<td>11.26</td>
</tr>
<tr>
<td>Tensor-based tracking method with GLCM contrast maps</td>
<td>0.9422</td>
<td>0.0189</td>
<td>10.53</td>
</tr>
</tbody>
</table>
the detected ones, particularly in inline 395–399. The mean of the SalSIM indices of the tracked boundaries is 0.9362, which is almost the same as those of the detected boundaries, 0.9348. It shows that the proposed tracking method is robust and efficient enough to delineate salt-dome boundaries.

Influence of noise on proposed tracking method

Because in some cases, the noise level in the reference section may be different from those of predicted sections, to eliminate the influence of noise on the tracked process, we propose the noise-adjusted tracking method, which selects PCs corresponding to the greatest S/N other than the greatest variation. In the local volume, we manually add Gaussian noise with different variances to seismic sections. In the reference section inline 400, the added zero-mean Gaussian noise has a variance of 0.01. In contrast, the noise level increases to 0.05 in predicted sections. We use the manually labeled reference boundary to synthesize tracked boundaries in inline 389–409. In Figure 23, we compare boundaries synthesized by the tensor-based tracking method with or without noise adjustment. With the increasing offset between the reference and predicted sections, the appearance of noise magnifies the error in the localization of tracked points. Because the noise-adjusted tracking method is robust to noise, the difference of two curves in Figure 23 grows with the increasing offset. The tracked boundaries obtained by the former one have a mean of SalSIM indices, 0.9468, which is greater than that of the boundaries synthesized by the latter one, 0.9396. It shows that the proposed noise-adjusted tracking method can alleviate the influence of noise and enhance the robustness of the tracking process.

CONCLUSION

In this paper, we propose noise-robust algorithms for detection and tracking of salt domes in postmigrated seismic volumes. By using a texture-based gradient and a 2D DFT-based dissimilarity measure, we are able to accomplish noise-robust salt-dome detection with high accuracy and excellent efficiency. Then, by tracking a detected boundary from an initial seismic section through a 3D volume, we further improve the efficiency of the labeling process. The tensor-based noise-adjusted subspace learning approach proved very useful for the tracking of salt-dome boundaries, especially when noise of varying levels is present. Combined together, the detection and tracking algorithms yield a practical salt-dome labeling system that is effective, efficient, and robust.

ACKNOWLEDGMENTS

This work is supported by the Center for Energy and Geo Processing at Georgia Tech and King Fahd University of Petroleum and Minerals.
REFERENCES


