The Seismic Analyzer: Interpreting and Illustrating 2D Seismic Data

Daniel Patel, Christopher Giertsen, John Thurmond, John Gjelberg, and M. Eduard Gröller, *Member, IEEE*

Abstract—We present a toolbox for quickly interpreting and illustrating 2D slices of seismic volumetric reflection data. Searching for oil and gas involves creating a structural overview of seismic reflection data to identify hydrocarbon reservoirs. We improve the search of seismic structures by precalculating the horizon structures of the seismic data prior to interpretation. We improve the annotation of seismic structures by applying novel illustrative rendering algorithms tailored to seismic data, such as deformed texturing and line and texture transfer functions. The illustrative rendering results in multi-attribute and scale invariant visualizations where features are represented clearly in both highly zoomed in and zoomed out views. Thumbnail views in combination with interactive appearance control allows for a quick overview of the data before detailed interpretation takes place. These techniques help reduce the work of seismic illustrators and interpreters.

Index Terms—Seismic interpretation, Illustrative rendering, Seismic attributes, Top-down interpretation

1 Introduction

Oil and gas are valuable resources accounting for around 64% of the total world energy consumption [10]. Oil and gas search and recovery is an economically valuable but complex task. Imaging the subsurface for exploration purposes is highly expensive. The imaging surveys consist of sending sound waves into the earth and recording and processing the echoes. Throughout the article we refer to this processed data as the seismic reflection data.

Due to the measuring expenses, an iterative approach for collecting data is taken. The first stage of a search typically involves collecting multiple 2D seismic slices which are analyzed by a team of geologists and geophysicists. If an area showing signs of hydrocarbons is discovered, 3D seismic reflection data is collected and analyzed. If further indications of hydrocarbon accumulation are found in the new data, drilling a well might be considered. Irrespective if the drilling hits a reservoir or not, it will give deeper insight into the data due to the process of bore hole logging. A bore hole log consists of physical measurements along the well path such as mineral conductivity, radioactivity and magnetism.

The current work flow in searching for oil and gas is to start a detailed interpretation of seismic structures. When enough structures have been interpreted to get an overview of the data, the results are discussed by an interdisciplinary team. This bottom-up approach is time consuming. The interpretation is challenging due to the low resolution and noisy nature of the seismic data. In cases of doubt, the interpreter often creates several alternatives of the same seismic structure. In addition, it is not uncommon that the team disagrees on the interpretation and decides that parts of the data must be reinterpreted. As soon as a consensus is achieved, the interpretation is documented for further dissemination outside the team. As part of the documentation, a seismic illustrator draws illustrations of the interpretation. Traditional illustrating is a time-intensive task and is therefore done late in the work flow.

In this work we present a top-down interpretation approach as a step prior to the currently used bottom-up interpretation. This alleviates some of the issues in the bottom-up approach. The top-down interpretation is performed with a sketching tool that supports coarse interpretation and quick creation of communication-friendly seismic illustrations. The inaccuracy in the seismic data and the need for preciseness during bottom-up interpretation can easily lead the interpreter to wrong conclusions based on insufficient information. When reinterpretation is required due to wrong conclusions, the long time lapses between interpretation and meetings is a big problem. With our top-down procedure, we facilitate short interpretation-to-meeting cycles. This enables earlier and more frequent discussions of interpretation hypotheses. Hypotheses can be clearly annotated, presented and possibly discarded earlier. A common understanding of the data can be achieved before the detailed interpretation starts. This can reduce the need for reinterpretations during the bottom-up approach. One can save time identifying which structures to focus on during the bottom-up interpretation. Also, no tools for quickly creating seismic illustrations exist. Currently general drawing software is used to illustrate seismic slices.

We focus on interpreting and illustrating 2D slices instead of 3D volumes for several reasons. Only 2D data exist in the earliest stage of data collection. Also it is common to do a slice by slice interpretation of the seismic volume. Furthermore, when creating illustrations for communication purposes, illustrators make heavy use of 2D slices. Even for 3D illustrations, cutouts are often used with 2D illustrated slices on the cutout side surfaces as seen in Figure 1.

1.1 Interpreting Seismic Data

Oil and gas is created when organic material is deposited and then buried, followed by the application of pressure and heat over a long period of time. The produced oil and gas will migrate upwards and accumulate in reservoir structures such as anticlines (seen in left Figure 1) or fault traps. Oil and gas is searched for by looking for signs of these depositional or reservoir structures. Due to the limited resolution of seismic data, one must work with interpretable events such as the horizons. Horizons delineate rocks with different mineral densities or porosity characteristics. The horizons can be seen as bright or dark lines in gray-level reflection data.

The existing bottom-up work flow at the Norwegian oil company StatoilHydro is to interpret about each 25th yz slice along the x-axis and then about each 25th xz slice along the y-axis. For each slice, typically around 5 key horizons are identified and traced out. It takes about 2-3 hours to interpret one slice of a standard-sized dataset. Certain points at a horizon might have several continuation options. In these cases, each option is interpreted and stored as a separate horizon. Deciding which horizon is correct, and which is not, is taken later when a better overview exists. When all slices are interpreted, different methods of automatic growing and surface interpolation are performed to complete the horizons in the slices that were jumped over. After an in-

- *Daniel Patel is with Christian Michelsen Research, Bergen, Norway, E-mail: daniel@cmr.no.*
- *Christopher Giertsen is with Christian Michelsen Research, Bergen, Norway, E-mail: chrisgie@cmr.no.*
- *John Thurmond is with StatoilHydro, Bergen, Norway, E-mail: jothu@StatoilHydro.com.*
- *John Gjelberg is with StatoilHydro, Bergen, Norway, E-mail: johgje@StatoilHydro.com.*
- *Eduard Gröller is with Vienna University of Technology, Austria, E-mail: groeller@cg.tuwien.ac.at.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mailto:tvcg@computer.org.

1077-2626/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

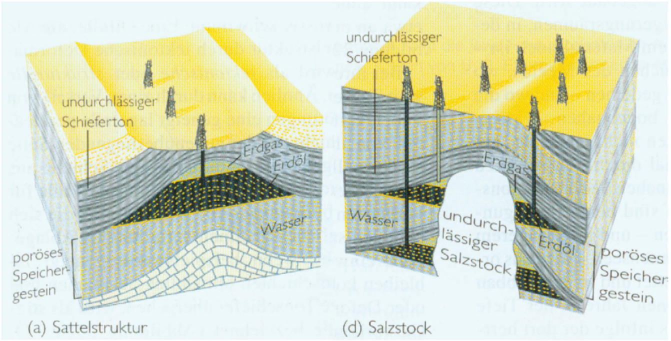

Fig. 1. Hand crafted illustrations of an oilfield. Left: Notice how the brick texture in the lowest layer and the stippled lines in the oil area are bending along the strata. Pictures are taken from Grotzinger et al. [8]

terpreter is satisfied with the horizons, a multidisciplinary discussion will take place to discuss the interpretation.

In addition to the horizons, the stratigraphy of the data is an important feature during interpretation. The term stratigraphy originates from the word stratum, meaning a layer of rock or soil with internally consistent characteristics that distinguishes it from contiguous layers. Strata can be found by either searching for areas with certain textures, or groups of horizons with common trends. This is referred to as seismic textures in literature [10]. Figure 2 gives some specific seismic examples. To support interpretation, different types of seismic textures in the seismic reflection data can be calculated automatically. Calculating the seismic texture *chaos* for a slice results in a new slice where the value of each pixel encodes the variance of the gradients in the pixel's neighbourhood (seen in bottom right Figure 3). The chaos attribute helps to indicate areas where horizons are difficult to track or to indicate stratigraphic regions containing salt (white area in right Figure 1). Such calculated slices are referred to as derived attributes. Other examples of derived attributes are reflection intensity and dip. Reflection intensity is the amplitude of the reflection strength in frequency space. Areas of strong reflections will have high reflection intensity. Dip is a measure of the angle of the seismic data in a neighbourhood by looking at the gradient values of the reflection data. See Figure 3 for some attribute examples.

In the domain of seismic interpretation, we facilitate visualization of derived attributes. We introduce the novel approach of identifying the horizons in the reflection data in a preprocessing step, and representing them as a collection of curves. We also present a filtering on the identified horizons so that they can be grouped together into strata according to similarity properties. Finally we enable the visual representation of well logs to be spread along horizons crossing the well.

1.2 Illustrating Seismic Data

We present novel illustrative visualization techniques of seismic data inspired by geological illustrations such as Figure 1. The patterns used in geological illustrations convey rich information. There exist over a hundred unique seismic patterns, each representing different types of rock-formation characteristics, also known as lithologies. The US Federal Geographic Data Committee [2] has defined a standard and a symbol lexicon for these patterns. The patterns are cleverly designed and constitute a taxonomy. Lithologies of the same subgroup are represented with similar patterns. Thus, even if a geologist does not recognize the exact lithological meaning of a pattern, he should be able to identify the general group it belongs to. Patterns or textures, and the way they bend are crucial information in geology. We explore the use of textures at early interpretation stages to easier express the interpreters inner model of the data.

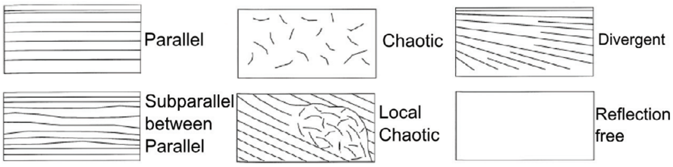

Fig. 2. Examples of stratigraphic patterns a seismic interpreter looks for during interpretation. Image is from Iske and Randen [10].

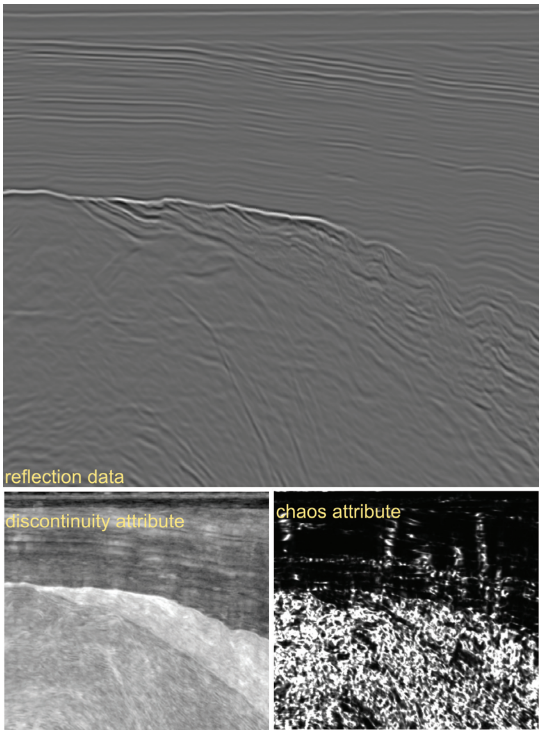

Fig. 3. Slices with reflection data at top and derived attributes below.

The data modalities that drive our illustrative process are the derived attributes, the traced horizons, and the well logs. Our illustrative rendering techniques map these data modalities to textures and lines that bend along the horizons in the seismic reflection data. To achieve the 'bending' of textures and lines, we calculate a 2D parameterization of the seismic reflection data. The mappings from data modalities to illustrative renderings are defined in terms of texture and line transfer functions. Each mapping of a modality results in what we call an illustrative layer. The rendering order of the illustrative layers defines which layers are in the background as context and which layers are in the foreground as focus. Different illustrative mapping techniques are defined for each seismic modality. Horizon data is either manually selected by user-picking or filtered according to horizon properties such as angles or average reflection strength. The selected horizons are then either mapped to textures with the texture transfer function or rendered as coloured lines using the line transfer function. For well logs, textures can be assigned to intervals of the well log values and spread out to a user-defined width along the horizons that cross the well. Finally, derived attribute data can be mapped by assigning intervals of the values to either textures with opacities, or to coloured lines with opacities. These rendering algorithms make it possible to merge all the data modalities into one multi-attribute illustrative image. For an intuitive understanding of the results, the mappings have to be carefully chosen. These techniques result in informative images that can be used directly as illustrations. This reduces the illustrator's time consuming manual work for tasks such as drawing deformed textures. An overview of the seismic analyzer can be seen in Figure 4.

In the domain of seismic illustrative rendering, the novelty of our approach lies in the parameterization of the seismic slice which allows for texturing, for line drawing and for dragging out textures from well logs. Also we introduce the concept of mapping seismic modal-

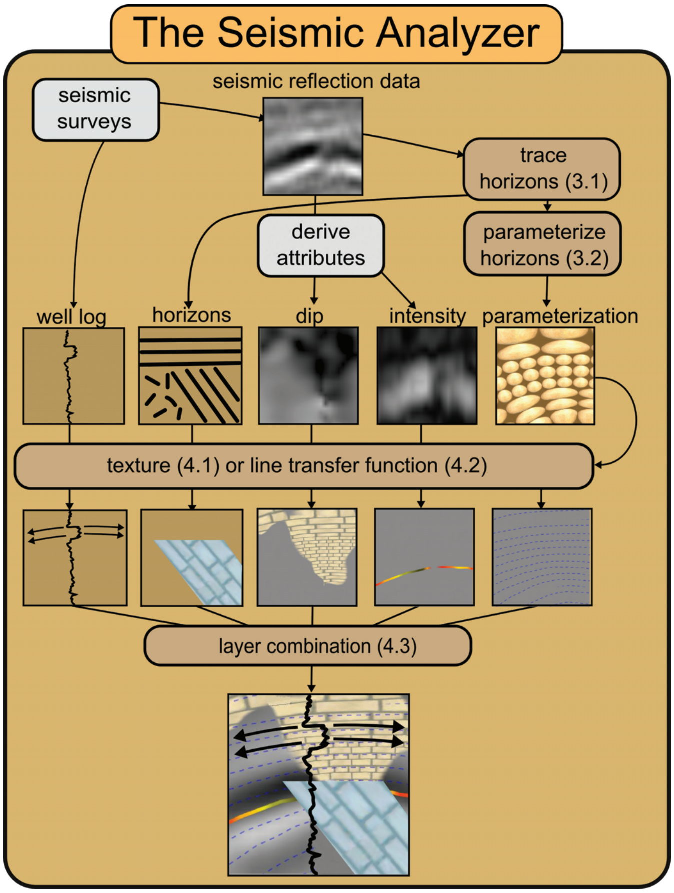

Fig. 4. The seismic analyzer. The brown rounded rectangles represent our algorithms and refer to the sections describing them. The 'seismic surveys' rectangle represents the process of obtaining the seismic data. The 'derive attributes' rectangle represents the process of deriving attributes using external software.

ities to illustrative layers which achieves multi-attribute visualization, illustrative visualization, and scale invariance.

2 Related Work

Several papers discuss processing and visualization algorithms for 3D seismic data. The visualization algorithms presented in papers and in commercial solutions are mostly direct volume rendering of the seismic data and surface rendering of interpreted objects such as horizons. In slice visualizations, horizons are represented as lines. For accurate structural interpretation, some papers deal with horizon extraction as in Castanie et al. [5] or fault extraction as in Jeong et al. [11] and Gibson et al. [7]. Pepper and Bejarano [15] give an overview of automatic interpretation methods. Plate et al. [16] and Castanie et al. [5] deal with handling large seismic volumes. Ropinski et al. [18] cover volume rendering of seismic data in VR. They present spherical and cubic cutouts which have a different transfer function than the surrounding volume. Commercial software used in oil companies include HydroVR [13] and Petrel [1]. None of these works deal with illustrative techniques or top-down interpretation as presented here. Papers that cover horizon extraction do it semi-automatically. The user has to select a seedpoint from where the horizon will be grown according to some user defined connectivity criteria. Our work differs in that we automatically pre-grow all horizons.

In our previous work [14] illustrative techniques for seismic data was also presented. That work dealt with the presentation and validation of *interpreted* seismic data. In this work we introduce a toolbox for interpreting and illustrating *non-interpreted* seismic data. This provides stronger illustrating capabilities than in our previous work. The parameterization in the previous work [14] required manually interpreted *complete* horizons. Now we introduce a parameterization that works directly on uninterpreted data by accepting automatically created horizon *patches*. We extend the texture transfer function from the previous work by introducing a flexible GUI and layered illustrative transfer functions for lines, wells and the automatically extracted horizons. Furthermore we propose how these techniques can be used in concert to improve the current work flow in oil companies by introducing the concept of top-down interpretation.

The 2D parameterization we present in this paper differs from parameterization methods used for texturing in other domains. 2D textures are applied to 2D images in both vector field visualization, such as Taponecco et al. [19], and in brush stroke synthesis as in Hays and Essa [9]. Our method differs in that we calculate the texture bending so it follows structures specific to seismic reflection data and not general gradient trends or edges. How we create the 2D parameterization is also different. Taponecco et al. [19] create 1D lines that are parameterized in length and then expanded in thickness to create a 2D parameterization. This procedure is performed locally over a collection of evenly distributed lines that cover the 2D image. This however results in overlapping textures. Our method considers the 2D space globally to create a complete and non-overlapping parameterization.

Much work has been done in multi-attribute visualization, Bürger et al. [4] present a state of the art overview. Crawfis and Allison [6] present a general framework where textures, bump maps and contour lines are used for multi-attribute visualization. Kirby et al. [12] present multi-attribute visualization of 2D flows using concepts from painting. They visualize flow attributes using procedural glyphs on a colour-coded background. Taylor [20] takes a general approach by combining colour, transparency, contour lines, textures and spot noise. He succeeds in visualizing four attributes simultaneously. However, little work has been done in multi-attribute visualization of seismic data.

3 Extracting Horizons and Parameterizing the Reflection Data

In this section we describe how the horizon structures are found and how the 2D parameterization is calculated. The horizon lines are used directly in the visualization of the data and as input to the 2D parameterization. The horizons and the parameterization is calculated in a preprocessing step prior to the visualization.

3.1 Tracing the Horizons

We adapt the method described in Iske and Randen [10] to trace out horizons. By considering the seismic reflection data as a height field, the horizons are running along valleys and ridges of the height field. We automatically trace out some of these valleys and ridges. Our method differs from existing horizon tracing algorithms since it does not require a user defined seed point for each trace. We go through all samples in the seismic slice and create traces for samples that are local maxima or minima in a vertical neighbourhood of 3 samples. The result is a collection of lines going through the horizons of the slice.

3.2 Parameterization of the Horizons

To achieve the effect of textures and lines following the orientation trend of the underlying reflection data, we create a parameterization from the traced horizons. The parameterization creates a relationship between image space and parameter space as seen in Figure 5a and b. There are four steps in determining the parameterization. The first step is to create suggestive line segments that indicate the horizons in the reflection data. We use the extracted horizon lines for this. The second step is to calculate the vertical v parameter values from the horizons. Thirdly, from the v parameter values, the horizontal u parameter values are calculated. Finally the u parameter values are normalized to minimize distortion in the parameterization. The parameterization process will ensure that horizons are mapped to straight lines in parameter space. Inversely, this guarantees that straight illustrative textures and

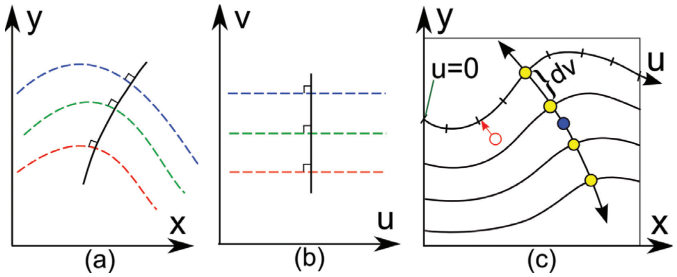

Fig. 5. Relationship between image space (a), and parameter space (b). In (c) is shown the procedure of finding the *u* parameter for a point (red circle) in image space by tracing from the point along a curve normal to the *v* parameter until it hits a *u* parameterized *v*-isocurve.

straight lines in parameter space will be aligned with the horizons in image space.

In the second step we calculate the *v* parameter by sweeping a vertical line from left to right over the horizons (Red line in Figure 6). The sweep line consists of a set of control points with unique *v* values. Initially there is one control point at the bottom of the line with *v*=0 and one at the top with *v*=1. The *v* values between control points are linearly interpolated. As the sweep line moves to the right, it will intersect the horizon lines. At the position on the sweep line where it intersects the start of a horizon, a new control point is created (double circle) which is assigned the interpolated *v* value at that point on the sweep line. For the following intersections of the same horizon line, the associated control point will update its position according to the intersection but will keep its initially given *v* value. One can imagine the sweep line as a rubber ribbon getting hooked on and off the horizons. After all points in the slice have been assigned *v* values, a 2D smoothing is performed to smooth out the discontinuities that arise just behind the horizon ends (see Figure).

Finding the *u* parameterization involves finding a mapping from vertical lines in parameter space (vertical black line in Figure 5b) to image space (Figure 5a). We want the parameterization to be angle preserving so that for instance the 90 degree angles at the edges of bricks in a brick texture are more or less preserved when mapped to image space. For this to be fulfilled, we require that the vertical lines in parameter space are always normal to the *v*-isocurves. We find an initial vertical line in parameter space by tracing two lines from the middle of the image space (blue dot in Figure 5c), one in the normal direction of the *v* parameterization, and one against the normal direction. This line is then parameterized according to its curve length and is divided into intervals of length dv (yellow dots in Figure 5c). From each of the interval ends (yellow dots) we span out *v*-isocurves. Each *v*-isocurve is *u*-parameterized according to its curve length and is set to 0 at the intersection with the left image border. The image space has now been divided into strips. Finally, the *u* value for any point in a strip is found by tracing from the point's position in the *v* gradient direction until a *v*-isocurve is hit (red line in Figure 5c). The point's *u* value is set to the *u* value at the isocurve intersection.

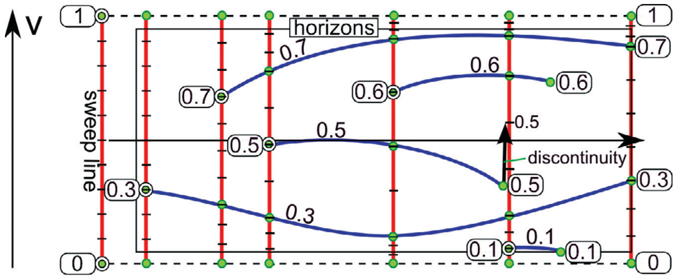

Fig. 6. Calculating the *v* parameterization by sweeping the red line with green control points from left to right. The blue lines are the horizons. The numbers are the *v* parameter values of the control points. Values in between the control points of a line are linearly interpolated.

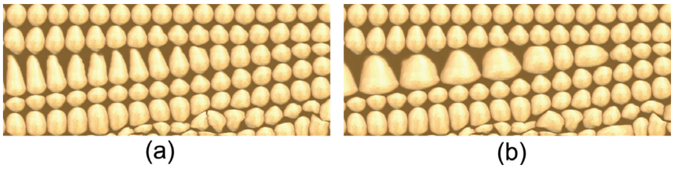

Fig. 7. Part of the reflection data textured with a ball texture to present the parameterization. Before (a) and after (b) isotropic correction.

We illustrate the resulting parameterization with a ball texture in Figure 7a. Each row of balls represents a parameter strip. One can see that the balls in the third row from the top become stretched and anisotropic as the strip's upper and lower *v*-isocurves diverge. We correct the parameterization so that the *u*/*v* ratio is constant as can be seen in the right image. This is done by remapping the *u* parameter for each strip so that it increments along the curve length relative to the thickness of the strip at that point. This ensures that textures are drawn with a consistent width/height ratio. The texture size however does vary. Varying texture sizes can be useful information during seismic interpretation since they communicate the degree of divergence of the horizons in an area.

4 Illustrative Rendering of Seismic Modalities

We present visualization techniques that filter and map the seismic modalities to visual representations which are more intuitive to understand than their direct representations. Combining filtering and mapping of the data to a visual representation has several advantages. Firstly, often only certain intervals in the value range of a modality are of interest to show. Uninteresting value ranges can be set to have no visual representation and value ranges of interest can be mapped to prominent visualizations. Secondly, scale invariance is achieved by changing the sparseness of the visual representation. By this we achieve that the image space is neither underloaded nor overloaded with visual information no matter how small or large the slice is. This will be discussed in more detail in section 5.1. Our approach also allows for multi-attribute and focus+context visualization where one can apply special rendering styles for modalities and value intervals that are of particular interest. In cases where the visual representations of several illustrative layers overlap, one can define a higher importance for one illustrative layer by rendering it on top of the others.

We have defined two generic techniques for mapping a seismic modality to an illustrative layer, i.e. texture transfer functions and line transfer functions. These two techniques define the data filtering and representation assignment for the seismic modalities. Some modalities have additional parameters that determine their appearance. By manipulating the texture and line transfer functions, the domain expert has a high degree of freedom to visualize and explore the multi-attribute data in real time. A texture transfer function assigns opacities and textures to a value range whereas a line transfer function assigns opacities and coloured lines to a value range. In effect the opacity assignment defines the data filtering, and the texture or colour assignment defines the representation mapping.

4.1 The Texture Transfer Function

The texture transfer function maps the scalars of a modality to an opacity and to a texture. The opacity is defined by a graph along the scalar axis. The scalar-to-texture mapping is described by discretely positioned texture references along the scalar axis. Scalars between two texture references will be represented by a weighted blend of the adjacent textures. The weighting is defined by the relative distance to the texture references. Examples of texture transfer functions defined in our GUI can be seen on the left side of Figure 8.

The textures are mapped to image space by a 2D parameterization. Either the parameterization of the reflection data or a basic uniform

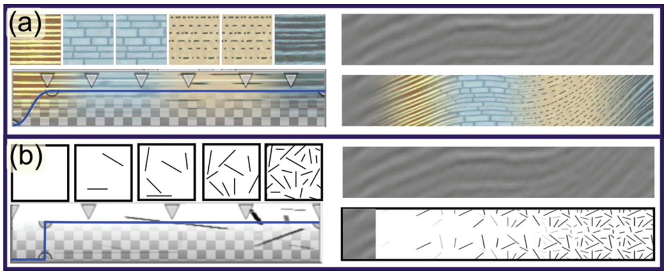

Fig. 8. Left: Texture transfer functions with blue graphs defining opacities. (a) Right: Textures following the parameterization of the reflection data. (b) Right: Textures following a uniform parameterization. Texture lookup values increase linearly from left to right in (a) and (b).

axis aligned parameterization which is independent of the horizon parameterization, is used. The latter parameterization is best suited for visualizing areas which have poorly defined horizons. Examples are areas that are chaotic, have weak or no reflections or that contain the geologic discontinuities called faults. Figure 8a and 8b show the two different types of parameterization. For each illustrative layer, multiplicative factors of the horizontal and vertical repeat rate of the textures must be defined by the user. We use the texture transfer function to control the visualization of the horizons, the well logs and the derived attribute slices.

For the traced horizon lines, we calculate measures like length, strength and angle. Each horizon line has a segmentation mask around it where the texturing will take place. The user can decide which horizon measure the transfer function will use. An example of a horizon transfer function is seen in Figure 9b. It is also possible for the user to pick, with the mouse, a subset of the horizons to apply the texture transfer function on.

Well logs, being physical measurements along a vertical line in a slice, are represented by assigning textures and opacities to the well log values. The textures are spread out horizontally from the well, along the crossing horizons, for a user defined distance. For well logs, the 2D parameterization is used both for the texture parameterization and for ensuring that the textures move along the horizons outwards from the well log. See Figure 10 for examples.

The well log can be used in a depth mode to define textures that vary as a function of the well depth. This is achieved by using a synthetic well log with values that increase linearly with the depth. In this mode, the opacity of the texture transfer function defines where to texture along the depth, and the texture assignment defines which textures to use along the depth. We refer to this as a depth transfer function. In the depth mode, the user can also move the well horizontally to perform the depth varying texturing at a location that intersects the stratum that is to be texturized. The use case in Section 5.2 will give examples of depth transfer functions.

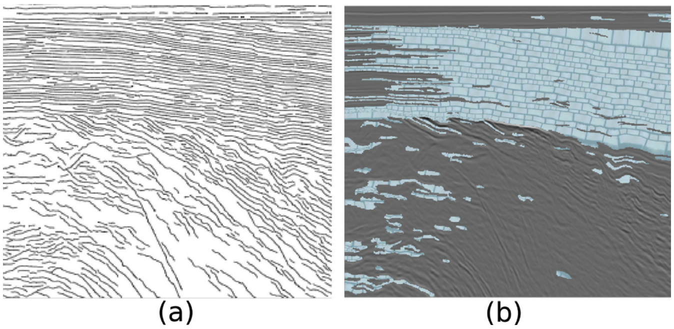

Fig. 9. a) All extracted horizon lines. b) Horizons with angles between 2 and 10 degrees are textured with a brick texture. The original seismic reflection data is shown in Figure 3.

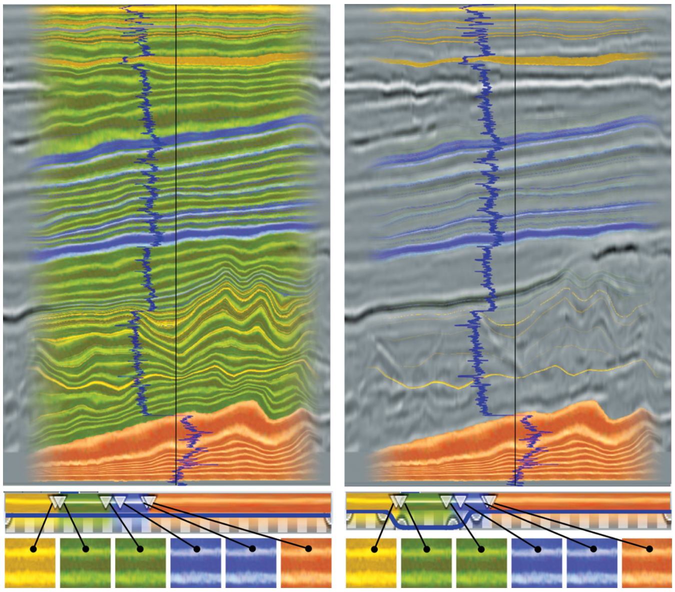

Fig. 10. Two well log transfer functions. Left: a well log with full opacity. Right: a well log with full opacity only for low and high well log values. The vertical line shows the well log path. The blue vertical graph shows the values of the well log for the well's gamma-ray radioactivity.

For derived attribute slices, the texture transfer function maps the scalar values to textures and opacities. Transparency, except for in the transitions into and out of intervals of interest, will create halos around the area of interest (Figure 11). This has the added effect that the halo thickness suggests the gradient magnitude of the attribute where thicker halos indicate smaller gradients. In general, texture transfer functions allow the interior, the transition, and the exterior areas of the values of interest to be shown in different ways, with the possibility to see underlying data.

4.2 The Line Transfer Function

An illustrative layer can also be created by using a line transfer function. The line transfer function defines lines that are curved according to the 2D parameterization of the reflection data. The colours and opacities of these lines can be linked to any derived attribute. This enables lines to describe a derived attribute by disappearing, reappearing, and changing colours. Three further line appearance parameters can be controlled globally for an illustrative layer. They are the density of the lines, the thickness of each line, and the lines' stipple repeat rate. The line density defines the minimum distance between two lines in image space. In the bottom image number 7 of Figure 12, two line layers are shown, one with blue stippled lines and one with a line partially coloured in red, yellow and black. The blue lines show the angular trend of the reflection data and they are based solely on the 2D parameterization with no relations to a seismic modality. We refer to such lines as streamlines. The opacity for the other line layer is set to transparent for low reflection intensity values. This results in one single line going through an area of high reflection intensity. The varying colours of the line arise from its line transfer function. The different zoom levels in Figure 12 show line layers with varying density and line stipple settings.

In the same way as the horizons can be assigned textures using a texture transfer function, we can assign colours and opacities to the horizon lines using a line transfer function. The user decides which horizon measure will be used by the line transfer function. Visualizing horizon lines is a method commonly used by seismic illustrators to represent the trends in the reflection data in a sparse and illustrative manner. With our approach, an illustrator can use filtering and horizon selection to draw the lines, such as seen in Figure 9a, as opposed to tracing them out manually.

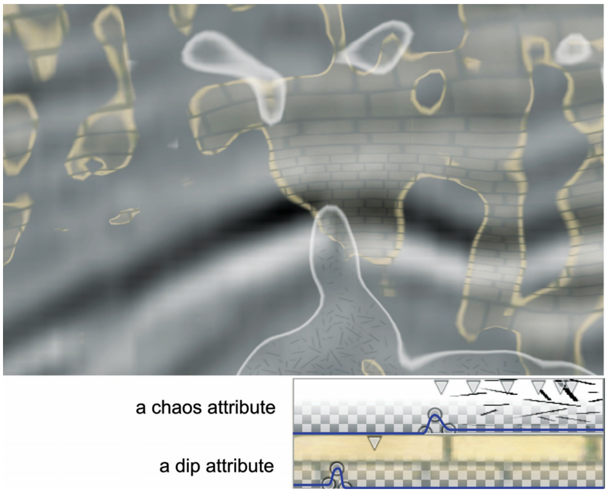

Fig. 11. The transfer functions for the derived chaos and dip attributes are defined at the bottom. The chaos attribute is transparent for low values and semi-transparent for high values with an opaque peak in between. The peak creates an opaque halo which separates low and high chaos values. A similar effect is seen for the dip attribute. The original reflection data is seen in the background.

4.3 Combining Illustrative Layers

The illustrative layers are combined into a resulting illustrative image by compositing them back to front using the over operator as described by Porter and Duff [17]. Any of the illustrative layers can be turned off to reveal the underlying layers. The user can choose the order of the layers and put the layer of highest importance in front so it is not visually obstructed by any other layer. In standard seismic illustrations, as opposed to our images, the reflectance data is not visible. We propose to integrate illustrative rendering with interpretation, therefore we enable showing the reflectance data, or other derived attributes, in the back most layer for comparison and verification reasons.

5 Results

Preprocessing the data for finding the horizons, using unoptimized Matlab code, takes from 10 to 20 minutes for a slice of size 500 by 500 samples. The parameterization takes less than a minute to calculate. The toolbox is implemented in Volumeshop [3]. Rendering requires little processing and is fast even on low end graphics cards.

In section 5.1 we present an outline of how our methods can facilitate a top-down approach for interpretation. Section 5.2 describes a use case highlighting the sketching capabilities of our tool.

5.1 Use Case: Top-Down Interpretation

Typically interpretation is performed in a time consuming bottom-up fashion with focus on details by looking at the reflection data on a fine scale. With our methods it is possible to first perform a coarse top-down interpretation. In the case a seismic survey lacks potential, a top-down approach allows for termination of the search at an early stage. This can happen as soon as a sufficient level of understanding is gained to draw conclusions. In the case of a promising survey, the top-down approach is also advantageous. At any stage, the interpretation at the current level can be used for communication purposes.

The data used in this example is a seismic reflection slice and the derived attributes chaos, dip and reflection intensity. The example is presented in Figure 12 and consists of three zoom levels. The first step in our approach is to visualize the data highly zoomed out. This gives an overview where one can identify interesting areas to investigate closer. Five zoomed out thumbnails can be seen in Figure 12, numbered one to five. The first image shows the gray reflection slice highly reduced where practically no information is left. This indicates that zoomed out overviews in the typical non-illustrative approach are not particularly useful. To get an impression of the seismic structures, we add an illustrative layer with sparse streamlines. In thumbnail image 2, the blue lines hint that the upper part of the seismic data is rather horizontal, the middle part is horizontal at the right side and angled at the left, and that the lower part is rather indecisive. For thumbnail image 3, we add another line layer with red lines in areas of high reflection intensity. Two areas with high reflection show up. We then add a brown brick texture layer in thumbnail 4 in areas with near horizontal dip. The top part of the seismic data stands out as brown. This area can be identified as a stratum with parallel seismic texture (see Figure 2). For thumbnail 5 we add another texture layer for showing chaotic areas. Two areas show up, one at the bottom and one at the left just below the parallel stratum. We now have an overview of the trends in the seismic reflection data and are ready to get a more detailed view. We zoom in on an area with low chaos and strong reflection values in image 6. At this detail level we manually adjust the texture repeat rate and the line density to get the appropriate detail level in the visualization. In image 6 one can clearly see the areas of chaos, the top stratum, and strong reflection values. Finally, in image 7, we decide to zoom in on an area with a flat spot. Flat spots are defined as areas of high reflection intensity and might indicate hydrocarbons. At this level we increase the blue line-density and render the lines in a sparse stippled style. We also increase the repeat rate for the textures further. To look closer at the variation of the reflection intensity, we add two

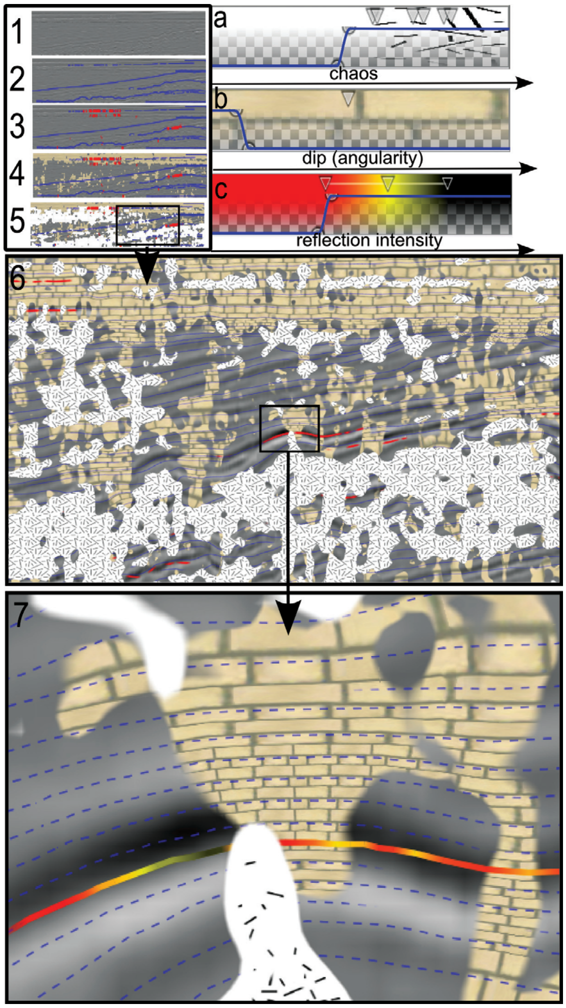

Fig. 12. A use case of an iterative drill down into the seismic reflection data. To get an overview, visual parameters are edited while looking at thumbnail-sized slices (1-5). This is followed by zooming into the data twice (5-6 and 6-7). a) is a texture transfer function on the derived chaos attribute, b) is a texture transfer function on the derived dip attribute, and c) is a line transfer function on the derived reflection intensity attribute.

more colours to the line transfer function. At this level one can see the different degrees of chaos by investigating the line density in the chaos texture. The interpreter can adjust the visualization, using the transfer functions and the layer ordering, so that it most closely matches his internal understanding of the data. By saving the visualization, the interpreter is able to externalize his gained internal understanding into an illustrative image that can be used as documentation. This example took about 10 minutes to interactively drill through. In this section we described how our methods support performing top-down interpretation as opposed to the existing bottom-up interpretation.

5.2 Use Case: Annotating Seismic Strata

We present a use case for interpreting stratigraphic layering in the search for hydrocarbons. This use case is created in cooperation with StatoilHydro who released the seismic data and the derived attributes to test our system. The survey had already been interpreted by the oil company but no interpretation information was given to us. Also the seismic interpreter, and the seismic illustrator, both from StatoilHydro, involved in this use case had little or no knowledge of this interpretation. The reflection data for this study is given in Figure 3. To get an initial high level overview of the data, the different strata are identified. The lithologies of the strata are unknown. The interpreter suggests the stratification as shown in Figure 13a. In the middle of the image a strong reflector is visually identified. It is thought to be a horizon separating two strata. While following the horizon from left to right the horizon splits up (yellow circle) and gives rise to two possible continuations seen as stippled lines in Figure 13a. The interpreter is uncertain whether the area between the stippled lines belongs to stratum 3 or to stratum 4. With our tool, the two stratification alternatives are sketched for the purpose of discussing them. Also, as more knowledge is gained, the textures defining the lithologies are changed.

At first, an illustrative layer is created to represent the bottom stratum 5. The stratum has a chaotic texture and it contains weak reflections due to its depth and due to strong reflectors above it. The interpreter also notices that the stratum contains reflection artifacts and concludes that the information there is not reliable. Using the parameterization of the reflection data in this area would be inappropriate since the horizons are not reliable there. Therefore, a uniform parameterization and a texture with chaotic lines is used to represent the stratum. To capture the stratum region, a depth transfer function is applied along a vertical line through the center of the image. The transfer function is set to transparent except in the depth interval where the vertical line intersects the stratum. The resulting region matches well with the interpreter's separation line seen in Figure 13b

A new illustrative layer is created and textures following the reflection parameterization are assigned to strata 1 to 4 by again using a depth transfer function. The result seen in Figure 13c matches well with the manually drawn strata lines in Figure 13a. However it was not possible to make the depth transfer function separate out stratum 2. The horizons in stratum 2 are well defined. Therefore stratum 2 is annotated by selecting its horizons by mouse picking and assigning a texture to them using a horizon transfer function. The result is seen in Figure 13d and identifies stratum 2 well. Now, one of the two alternatives of the sketch in Figure 13a is reproduced. The other alternative is quickly derived from the first alternative by moving the end-depth of the yellow texture and start-depth of the blue texture on the transfer function slightly lower (Figure 13e). The alternatives in Figure 13d and e can now be discussed among the experts. It is noticed that the fourth stratum has a somewhat distinct seismic texture (see section 1.1 for a discussion of seismic texture). It is decided to derive the discontinuity attribute with external software to see if it highlights stratum 4. If it does, then the membership of the undecided area can be resolved by comparing the discontinuity attribute to that of stratum 4. The attribute is depicted in Figure 3. By looking at the attribute one can see a distinct region in the middle. To compare this region with the annotated strata, we create a new illustrative layer with an attribute transfer function on the discontinuity attribute. Texturing only values of high discontinuity and overlaying the texturing on the illustrative layers in Figure 13e yields Figure 13f. The new illustrative layer overlaps closely with stratum 4 except for some random patches. It does not cover the undecided region. It is concluded that Figure 13e is correct since the region in question is now assumed to belong to stratum 3 but not to stratum 4. Based on the current stratigraphic mapping and due to the the strong reflection property of the horizon between stratum 3 and 4, a hypothesis is formed that stratum 4 consists of limestone. Therefore, in Figure 13g, the texturing of the fourth stratum is changed to a geological texture denoting limestone. Scrutinizing the reflection data of the stratum reveals mound structures (similar to the stratigraphic pattern 'Local Chaotic' in Figure 2). It is further hypothesized that the mounds might indicate karst bodies. Karst bodies are hollow structures created by reactions between carbonate rock and water that may act as hydrocarbon traps. An attribute is derived which is sensitive to mound like structures. To show these structures, a new illustrative layer is made. Attribute regions of high mound characteristics are displayed with one texture, and areas of medium mound characteristics are displayed with another texture (see Figure 13g and h). Figure 13h shows a zoom-in on some karst bodies. Figure 13i shows the layers with semi-transparency and opaquely emphasized strata borders and karst bodies. The interpreter now decides that this is as far as he can go with the current knowledge of the survey. This process took about half an hour and has shown that the investigated area has potential and is worth a further exploration with a detailed bottom-up interpretation. It has also saved the time of unnecessary detailed bottom-up interpretation of the topmost stippled horizon. Our tool enabled discussing possible interpretations and to arrive to conclusions at an early stage. We have also shown that the time spent creating illustrations with our system is in the order of minutes. Manually drawing illustrations of comparable quality would be in the order of hours.

There are seismic areas such as faults or noisy regions where correct automatic horizon extraction is not possible. Since the parameterization depends on the extracted horizons, the texturing will fail in these areas. As seen, even human interpreters have problems finding correct horizons in difficult areas. To address this, our system can mark out areas where parameterization fails, by using uniform texturing as discussed in Section 4.1 on an attribute that is sensitive to the problematic areas. This approach was presented in Section 5.1 for chaos areas. Ideally, textures should be discontinuous across faults. This would require the unsolved task of automatic and accurate detection of the fault surfaces. However with our methods one can use a fault sensitive attribute (there exists robust ones) and an appropriate texture pattern to mark possible fault areas where normal texturing would fail.

6 Conclusions

We have presented a toolbox with novel interpretation and rendering algorithms. It supports fast seismic interpretation and fast creation of seismic illustrations. The toolbox offers illustrative visualization, scale invariant visualization, and multi-attribute visualization. Uninterpreted seismic data has high uncertainties and fits into our quick and coarse top-down approach. Afterward, the more accurate bottom-up method is applied when higher certainty in the data has been gained. We believe our toolbox will increase the efficiency of seismic illustrators by automating time consuming tasks such as texture creation and horizon drawing. These tasks are currently performed with general drawing programs. We have informally evaluated the usefulness of our approach in 2D. We plan to extend it so that the slice plane can be positioned arbitrarily in 3D at interactive frame rates, and to integrate this approach with standard 3D volume rendering.

References

[1] Petrel seismic interpretation software, schlumberger information solutions (sis).

[2] *Federal Geographic Data Committee, Digital Cartographic Standard for Geological Map Symbolization*. Document FGDC-STD-013-2006, 2006.

[3] S. Bruckner, I. Viola, and M. E. Gröller. Volumeshop: Interactive direct volume illustration. acm siggraph 2005 dvd proceedings (technical sketch). In *ACM Siggraph Technical Sketch)*, 2005.

[4] R. Bürger and H. Hauser. Visualization of multi-variate scientific data. In *EuroGraphics 2007 State of the Art Reports*, pages 117–134, 2007.

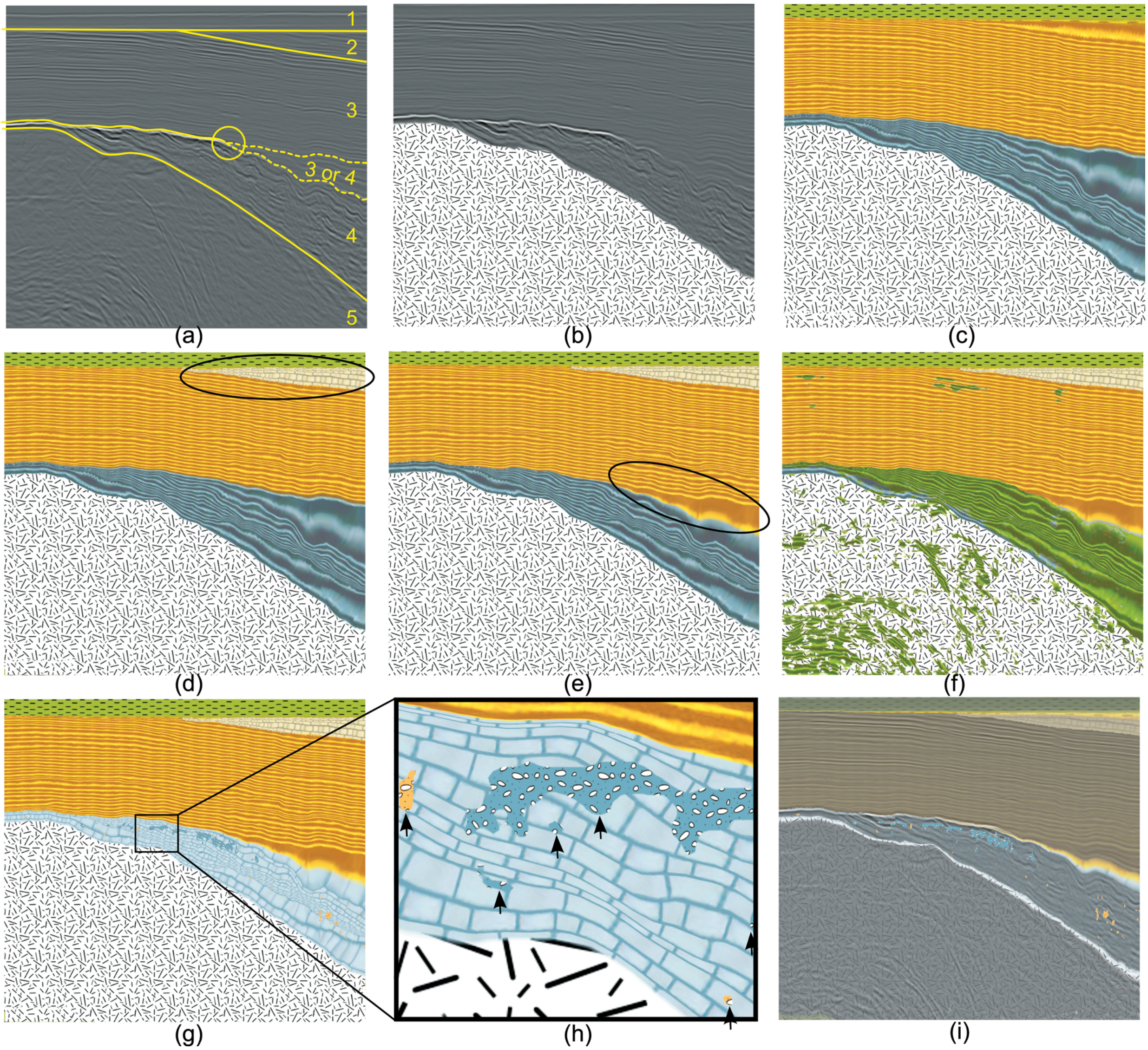

Fig. 13. Images created during an interpretation. Figure a) shows manually drawn yellow lines for separating strata. Figures b)-i) are rendered with our system. They are discussed in Section 5.2. Ellipses in (d) and (e) pinpoint the difference from the previous image. Arrows in (h) point out areas of medium (orange) and high (blue) mound characteristics.

[5] L. Castanie, B. Levy, and F. Bosquet. Volumeexplorer: Roaming large volumes to couple visualization and data processing for oil and gas exploration. *Proc. of IEEE Visualization '05*, pages 247–254, 2005.

[6] R. A. Crawfis and M. J. Allison. A scientific visualization synthesizer. In *VIS '91: Proceedings of the 2nd conference on Visualization '91*, pages 262–267, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[7] D. Gibson, M. Spann, J. Turner, and T. Wright. Fault surface detection in 3-d seismic data. *Geoscience and Remote Sensing*, 43(9):2094–2102, 2005.

[8] J. Grotzinger, T. H. Jordan, F. Press, and R. Siever. *Understanding Earth*. W. H. Freeman and Company, 1994.

[9] J. Hays and I. Essa. Image and video based painterly animation. In *NPAR '04: Proc. of the 3rd intn. symposium on Non-photorealistic animation and rendering*, pages 113–120, NY, USA, 2004. ACM.

[10] A. Iske and T. Randen, editors. *Atlas of 3D Seismic Attributes, Mathematics in Industry, Mathematical Methods and Modelling in Hydrocarbon Exploration and Production*. Springer, Berlin Heidelberg, 2006.

[11] W.-K. Jeong, R. Whitaker, and M. Dobin. Interactive 3d seismic fault detection on the graphics hardware. *Volume Graphics*, pages 111–118, 2006.

[12] R. M. Kirby, H. Marmanis, and D. H. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *IEEE Visualization '99*, pages 333–340, San Francisco, 1999.

[13] E. M. Lidal, T. Langeland, C. Giertsen, J. Grimsgaard, and R. Helland. A decade of increased oil recovery in virtual reality. *IEEE Computer Graphics and Applications*, 27(6):94–97, 2007.

[14] D. Patel, C. Giertsen, J. Thurmond, and M. E. Gröller. Illustrative rendering of seismic data. In H. S. Hendrik. Lensch, Bodo Rosenhahn, editor, *Proceedings of Vision Modeling and Visualization*, pages 13–22, 2007.

[15] R. Pepper and G. Bejarano. Advances in seismic fault interpretation automation. In *Search and Discovery Article 40170, Poster presentation at AAPG Annual Convention*, pages 19–22, 2005.

[16] J. Plate, M. Tirtasana, R. Carmona, and B. Fröhlich. Octreemizer: a hierarchical approach for interactive roaming through very large volumes. *Proc. of VISSYM '02*, pages 53–64, 2002.

[17] T. Porter and T. Duff. Compositing digital images. *Computer Graphics Volume 18, Number 3*, pages 253–259, 1984.

[18] T. Ropinski, F. Steinicke, and K. H. Hinrichs. Visual exploration of seismic volume datasets. *Journal Proc. of WSCG '06*, 14:73–80, 2006.

[19] F. Taponecco, T. Urness, and V. Interrante. Directional enhancement in texture-based vector field visualization. In *GRAPHITE '06*, pages 197–204, New York, NY, USA, 2006. ACM.

[20] R. M. Taylor. Visualizing multiple fields on the same surface. *IEEE Computer Graphics and Applications*, 22(3):6–10, 2002.