



ELSEVIER

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Knowledge Assisted Visualization

Knowledge-assisted visualization of seismic data

Daniel Patel^{a,b,*}, Øyvind Sture^a, Helwig Hauser^a, Christopher Giertsen^b, M. Eduard Gröller^{c,a}^a University of Bergen, Norway^b Christian Michelsen Research, Bergen, Norway^c Vienna University of Technology, Austria

ARTICLE INFO

Article history:

Received 19 April 2009

Received in revised form

14 June 2009

Accepted 26 June 2009

Keywords:

Knowledge-assisted visualization

Illustrative visualization

Seismic interpretation

Rapid interpretation

2D textures

3D textures

ABSTRACT

We present novel techniques for knowledge-assisted annotation and computer-assisted interpretation of seismic data for oil and gas exploration. We describe the existing procedure for oil and gas search which consists of manually extracting information from seismic data and then aggregating it into knowledge in a detail-oriented bottom-up approach. We then point out the weaknesses of this approach and propose how to improve on it by introducing a holistic computer-assisted top-down approach intended as a preparation step enabling a quicker, more focused and accurate bottom-up interpretation. The top-down approach also enables early representations of hypotheses and knowledge using domain-specific textures for annotating the data. Finally we discuss how these annotations can be extended to 3D for volumetric annotations.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Whether we like it or not the world is dependent on energy. Oil and gas accounts for around 64% of the total world energy consumption (Iske and Randen [11]). Thus searching for and recovering these resources is important in today's society. In this paper we describe how oil and gas search is performed and we propose using knowledge-assisted visualization for improving it. There are several aspects of how the search, using seismic interpretation, is performed that makes it fit very naturally into the paradigm of knowledge-assisted visualization. A visual symbolic language for capturing knowledge has already been developed in the geosciences for interpretation. There is a high need for expressive visualizations due to large degrees of collaborative work during interpretation. Finally, large amounts of money can be saved by increasing accuracy and reducing interpretation time. We will describe in detail how these aspects of current interpretation enable knowledge-assisted visualization to accelerate the search of oil and gas.

2. Related work

The field of knowledge-assisted visualization has been described in the paper by Chen et al. [4]. We are using the meaning

of data, information, knowledge and the process of knowledge-assisted visualization as defined in their paper. The entities of data, information and knowledge represent, in the order they are listed, increasingly higher degrees of abstraction and understanding. A system supporting knowledge-assisted visualization contains mechanisms for externalizing the user's knowledge gained using the system and mechanisms for explicitly expressing the knowledge in a visualization. Such a system may also contain domain-specific reasoning algorithms that assists the user in gaining knowledge. Illustrative rendering is a nonphotorealistic visualization technique using the advantages of conveying information through illustrations. This technique is well suited for representing gained knowledge and thus fits into the knowledge-assisted visualization paradigm. A good introduction to illustrative visualization and how it fits into the workflow of gaining knowledge can be found in Rautek et al. [22] and in the tutorial by Viola et al. [25]. In Rautek et al. [22] they argue for using illustrative techniques earlier in the knowledge acquisition process instead of only at the end for presenting the results as it classically has been used for.

In our work we extensively use illustrative textures to convey knowledge. Many techniques on texturing have been published. Owada et al. [17] presented an interactive system for texturing arbitrary cuts through polygonal objects. The user defines the texture flow by specifying a direction field and a distance field on the cut. The 2D texture is generated from a 2D exemplar through texture synthesis. Their method is general and therefore requires user interaction to specify the texturing. In our work we calculate a parameterization up front so texturing can be achieved quickly

* Corresponding author at: University of Bergen, PB 7803, 5020 Bergen, Norway.
E-mail address: daniel@cmr.no (D. Patel).

and without the need for texture synthesis. Since we target a domain-specific problem, many of the parameters defining the visualization are known prior to rendering, and less user specification is required. Lu and Ebert [15,16] generated illustrative medical renderings by applying 2D textures sampled from illustrations on volumetric data. 3D textures are created by combining color information from the illustrations with 3D clinical volume data. The synthesized 3D textures are made tileable using Wang Cubes. They do not deal with deforming the textures to follow the underlying flow of the data. Dong and Clapworthy [7] presented a technique that achieves 3D texture synthesis following the texture orientation of 3D muscle data. Their algorithm has two steps. First they determine the texture orientation by looking at the gradient data of the volume combined with a direction limited Hough transform. Second they perform a volumetric texture synthesis based on the orientation data. We identify the texture flow by using domain-specific methods as will be discussed later in this paper. The texture synthesis of Dong and Clapworthy has the drawback of not working on textures with large interior variation as textures in geologic illustrations commonly have. Wang and Mueller [26] used 3D texture synthesis to achieve sub-resolution zooming into volumetric data. With 2D images of several zoom levels of a tissue, they synthesized 3D volume textures for each level and used constrained texture synthesis during zooming to blend smoothly between the levels. They address the issue of sub-resolution details but do not consider texture flow. Kopf et al. [13] created convincing 3D textures from 2D texture exemplars where they are also able to synthesize semitransparent 3D textures. The 3D textures are created by gradually optimizing the similarity in local neighborhoods between the synthesized solid texture and the exemplar. Global similarity is achieved by color histogram matching from the 2D exemplar. For synthesis control different 2D exemplars may be specified for each of the orthogonal views of the wished solid texture. Due to the stochastic nature of the synthesis, full control is not possible and unexpected 3D textures may be generated. Stochastically synthesized textures may have low similarity to the original exemplar and are not ideal for our use as texture recognizability is important in our approach.

Much work has been done in multiattribute visualization. Bürger et al. [2] presented a state of the art overview. Crawfis and Allison [6] presented a general framework where textures, bump maps and contour lines are used for multiattribute visualization. Kirby et al. [12] presented multiattribute visualization of 2D flows using concepts from painting. Taylor [24] took a general approach by combining color, transparency, contour lines, textures and spot noise. He succeeded in visualizing four attributes simultaneously. However, little work has been done in multiattribute visualization of seismic data as we focus on in our works.

Ropinski et al. [23] covered volume rendering of seismic data in VR. They presented spherical and cubic cutouts which have a different transfer function than the surrounding volume. We incorporate and extend this concept in our work for combining illustrative visualization with scientific volume visualization inside cutouts. Plate et al. [21] and Castanie et al. [3] discussed the handling of large seismic volumes. Commercial software used in oil companies includes HydroVR [14] and Petrel [1]. None of these works discuss illustrative techniques or top-down interpretation as presented here. Horizon interpretation algorithms are typically semiautomatic requiring detailed and time consuming user involvement. Pepper and Bejarano [20] gave an overview of computer-assisted interpretation algorithms for seismic data. Several growing algorithms exist. Castanie et al. [3] proposed user seeding followed by growing based on the local waveform of the seedpoint. Interpretation software [1] performs growing in voxels that have been thresholded or in extrema or zero crossings

of the local waveform. For horizon extraction algorithms the user typically sets growing parameters and seeds and grows until a satisfactory result is obtained. The parameters to be set are related to the internal growing/image processing algorithms and can be difficult to understand. One might argue that the user is given too much low-level control. In our system we aim at avoiding the need for parameter tweaking in early exploratory phases by offering automatically pre-grown horizon candidates. A similar idea was developed by Faraklioti and Petrou [9] but which results in fewer detected and more fragmented horizons due to the requirement of horizon planarity.

3. Overview

In Section 4 we first describe the current method of oil and gas search. We then point out the weaknesses of this approach and propose how to improve on it by introducing a holistic and sketch based top-down approach. The top-down approach is intended as a preparation step enabling a quicker, more focused and accurate bottom-up interpretation. It enables early representations of hypotheses and knowledge by providing domain-specific annotations of the data using textures. The section concludes with a comparison of the old and the proposed new approach. In Section 5 we provide examples of annotated seismic data used in the top-down approach. In Section 6 we present a taxonomy of texturing techniques that can be used for annotating seismic volumetric data and their associated advantages and disadvantages. The article is rounded up with conclusions in Section 7.

4. Bottom-up and top-down interpretation

In this section we describe the current bottom-up interpretation pipeline, introduce our new top-down methodology and explain some of the automated interpretation techniques enabling the top-down approach. Afterwards we compare the two approaches.

4.1. Bottom-up interpretation

Seismic volumetric reflection data are used when exploring a geological prospect. The data are gathered by sending sound waves into the ground and processing the echoes. In the collected seismic data, hydrocarbon traps are searched for by looking for certain structural signatures. The first step in the interpretation is to get a structural overview of the data by identifying horizons, which are separations between rock layers, and faults, which are cracks and discontinuities in the rock layers.

Currently for volumetric data, interpretation is performed by a single person, slice by slice. The domain expert is looking at the data at maximum resolution and in detail tracing out structures manually. For more details about the interpretation pipeline, see Iske and Randen [11] and Patel et al. [18]. We have exemplified the interpretation process in Fig. 1. At the top of Fig. 1a, one can see a specific seismic slice under interpretation. The interpreter traces out horizons h_1-h_6 and faults f_1, f_2 . Then as more insight is acquired, the interpreter tries to connect the identified horizon pieces across faults (f_1, f_2) and through noisy areas (stippled lines between h_5 and h_6). Pieces he thinks belong together are shown with the same color. The conceptual pyramid below the slice illustrates this aggregation of insight. At the lowest level in the tree inside the pyramid, horizons and faults are identified with no relation to each other. At consecutive higher levels, the horizons are grouped. The groupings are represented with common nodes, and their interaction with faults is identified. Horizon and fault

interactions are depicted as curved edges from fault nodes to horizon group-nodes. When moving upwards in the pyramid, several lower level structures are grouped into fewer higher level structures. The result is that the amount of data is reduced, which is indicated by the narrowing pyramid shape, while insight or knowledge increases. When an overview has been reached, represented by the single node at the top of the pyramid, an expert meeting is arranged. A correct interpretation requires expertise in several fields. Therefore the overview is discussed by an interdisciplinary team of geologists, geophysicists, well engineers and other domain experts. When high-level structures can be seen in relation to each other, errors in the interpretation are more easily identified. The discussions will in many cases conclude that parts of the structures or interrelations between structures are incorrectly classified. As a consequence the prospect must now undergo a time consuming reinterpretation (Fig. 1b). A problematic region is indicated with

an ellipse in the slice of Fig. 1a. The interpreter incorrectly connected horizons h_1 and h_3 across fault f_1 while they should be considered as separate horizons. Interpretation errors in lower levels can easily propagate to higher levels. Therefore a reinterpretation is made after ungrouping horizons h_1 and h_3 . A different color is assigned to the group of horizons connected to h_3 in Fig. 1b due to the ungrouping. When the multidisciplinary team agrees on the reinterpretation, a more accurate understanding of the seismic data has been gained and areas of particular interest with regard to hydrocarbon potential are identified. They are indicated with ellipses in the slices of Fig. 1b and c. The interpretation is then focused on this area and is finally sent to a seismic illustrator. The illustrator manually generates illustrations that capture the gained high-level knowledge in a standardized and easy to understand way. Making such illustrations is quite a time consuming task. The process just described is what we refer to as the bottom-up approach.

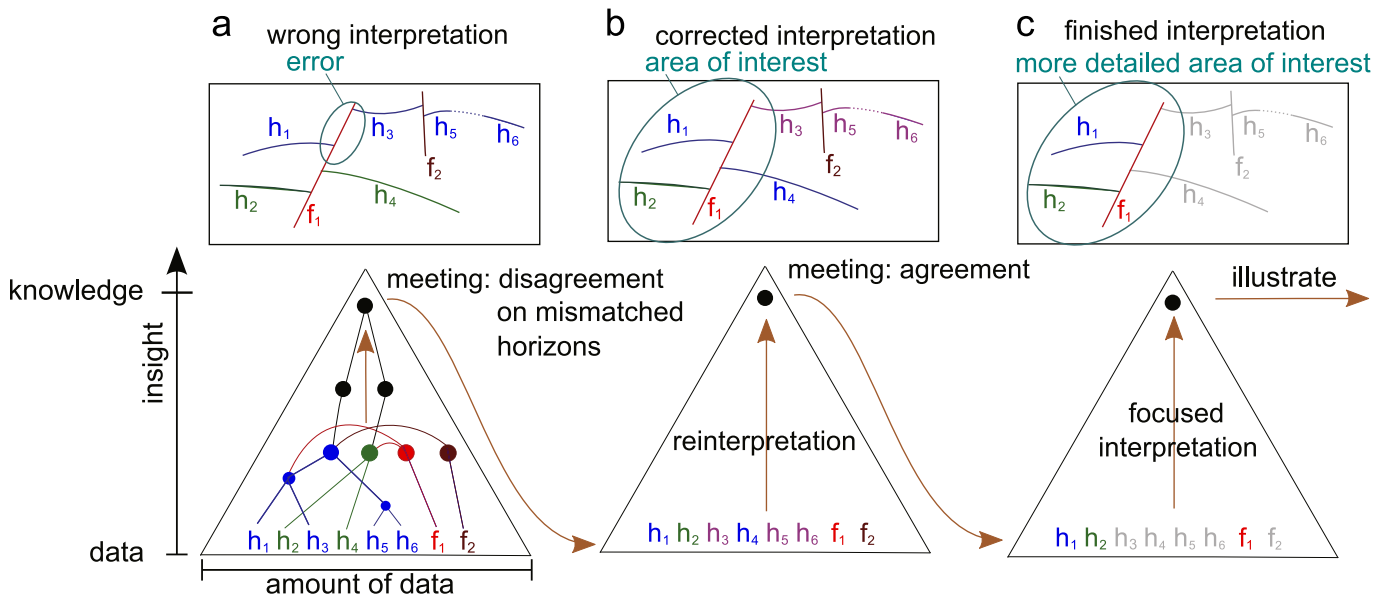


Fig. 1. An example of the steps performed during a bottom-up interpretation.

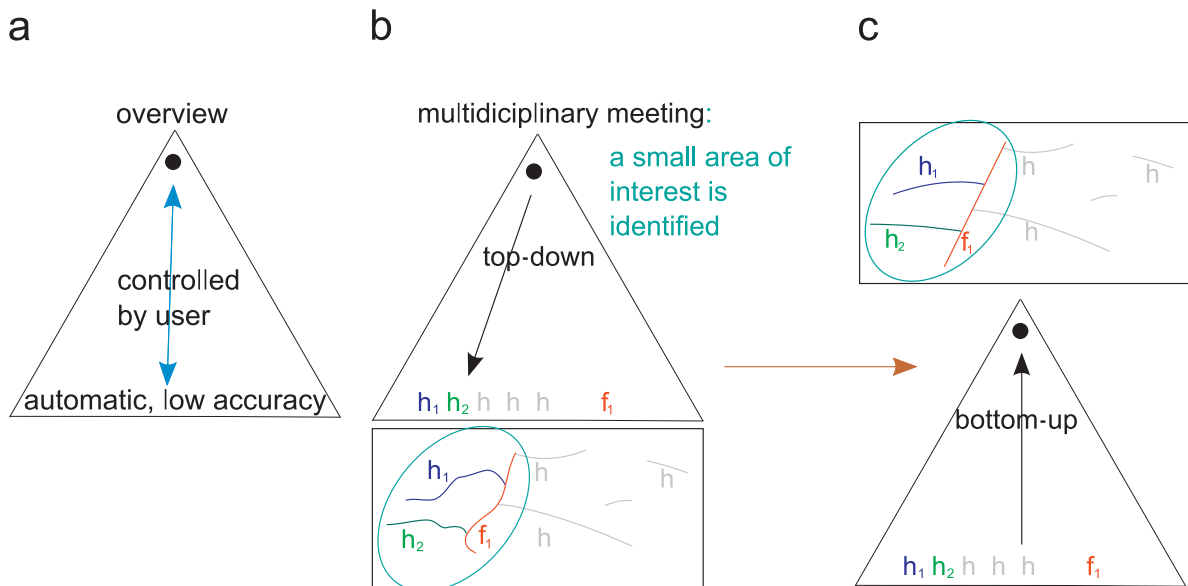


Fig. 2. The steps performed during a top-down followed by a bottom-up interpretation. An overview of the conceptual pyramid is given in (a). In (b) a top-down interpretation is performed prior to the bottom-up interpretation in (c).

The bottom-up approach does not scale well with the amount of data. With the rise of data sizes and resolutions due to technological advances, this approach is increasingly time consuming as it works on the highest level of resolution. Seismic volume data are inaccurate due to the complex and under-determined process of transforming the reflected sound waves into a 3D volume. Further inaccuracies stem from physical phenomena such as noise, multiple reflections and acoustic shadowing. Our top-down approach aims at improving the weak points of the bottom-up approach. Firstly, it is problematic to start with an *accurate and detailed* interpretation of uninterpreted seismic data due to their *inaccuracies and uncertainties*. Secondly, expert guidance from the multidisciplinary team comes late, after the time consuming detailed interpretation for the overview has been made. Due to the too late guidance, reinterpretations must often be performed. Ultimately, we wish to reduce the time necessary to create the final illustration and to allow for illustrations to be created at any stage of the interpretation.

4.2. Top-down interpretation

We propose to address the above mentioned problems by introducing a quick top-down interpretation stage at an early point in time. This is performed collaboratively by a multidisciplinary team of interpreters before the slower single person bottom-up analysis takes place (see Fig. 2). In the top-down stage, interpretation begins at a high level in the conceptual pyramid on a coarse level of detail by looking at the data with a highly zoomed out and abstracted view. The approach uses computer-assisted sketching and multiattribute visualizations for expressing and discussing hypotheses at an early stage during interpretation. To present hypotheses and knowledge, illustrative rendering is used. Illustrative techniques have been developed for the purpose of communicating high level aspects in a simple way. Elements of interest are illustratively emphasized to represent the knowledge acquired during the interpretation process. Illustrative methods allow for annotating the data so that the resulting image closely communicates the interpreters' internal models and hypotheses. This enables the interpretation team to clearly communicate and get a common understanding of each others ideas. When the understanding at the current level of detail is agreed on, a more detailed level can be investigated. A more detailed level is gained by either zooming in on a specific area of the data or by adding more data to the visualization by including new attributes. Since illustrative visualizations are created at all interpretation stages external communication outside the team is possible at any time. The bottom-up approach starts only after an agreed rough overview of the data has been made as seen in the pyramid of Fig. 2b. The gain of adding the top-down stage is to focus the bottom-up interpretation on important structures. This avoids interpreting areas of low or no interest, such as the unnecessarily interpreted gray areas in Fig. 1c. Furthermore uncertainty and the need for reinterpretation at late stages due to disagreements in the team is reduced. The top-down stage can also act as a screening to find out early if the prospect lacks potential hydrocarbon structures and should be abandoned.

In Fig. 2 the workflow of performing a top-down interpretation prior to the bottom-up interpretation is shown. We indicate the inaccuracies and sketchy nature of the top-down interpretation by using wiggly horizons and faults in the pyramid of Fig. 2b. These are concretized in pyramid Fig. 2c after a bottom-up interpretation confined to the identified area of interest was performed. Outside the area of interest, an even more sketchy and inaccurate interpretation was performed for overview reasons. Although the process is top-down from the user's perspective, it is bottom-up

from a computational perspective since it builds on the automatically extracted low-level structures at the bottom of the pyramid.

There are two mechanisms that enable top-down interpretation. The first is to relieve the user from performing time consuming low-level interpretation of structures. This is achieved by automatically preprocessing the data for identifying low-level structures. This mechanism is represented as the text at the bottom of the pyramid in Fig. 2a. The process can be seen as a computer-driven interpretation. However a computer-driven interpretation may generate inaccurate structures and interpretation suggestions as it cannot match the accuracy and experience of a human interpreter. The user must therefore be given the opportunity to quickly and easily browse through the computer-generated suggestions to select valid ones and aggregate them into higher level structures. This is the second mechanism, which is indicated by the blue arrow in the pyramid of Fig. 2a. The first mechanism is implemented by automatically deriving appropriate support data in a preprocessing stage before interpretation. The second mechanism is implemented by giving the user the control to browse the computer-generated support information and quickly create abstracted representations of the data. As opposed to raw data visualization, abstracted representations can be sparse, i.e. not cluttering and covering up the view. This yields two ways of aggregating data and getting overviews—by zoom-outs and by multiattribute renderings. Fig. 3 has examples of zoom-outs in the middle column and an example of multiattribute rendering is shown in the right image.

4.3. Automatic interpretation

Horizon candidates are automatically extracted in a preprocessing stage by adapting the method described in Iske and Randen [11]. Horizon candidates are identified by tracing out line segments following the valleys and ridges of the height field defined by the reflection values on a slice, see Fig. 4. The result is a collection of lines going through the horizons of the slice as seen in Fig. 4c. Our method differs from existing horizon tracing algorithms found in commercial software as it does not require a user defined seed point for each trace. This avoids having the interpretation flow interrupted due to setting seed parameters and waiting for growing results.

For each traced horizon line, we calculate measures like length, average strength and average angle. The user can filter horizons based on any of these measures. It is also possible for the user to select a subset of the horizons. Picked or filtered horizons can be visualized as line segments or texturing can be applied for annotation reasons. Each horizon line has defined a segmentation mask around it where texturing or coloring can take place. Transparency can be set so that the original seismic data will be visible underneath. The texture mapping is defined by a parameterization created from the extracted horizon lines. This is done to ensure consistent deformation and orientation of the textures with the underlying seismic data. For details about the parameterization, see Patel et al. [18].

4.4. Comparison

In Fig. 5a we compare the bottom-up and the top-down approach. The vertical axis represents the information level of the structures that have been identified in the interpretation. The black dot close to the origin represents the lowest level of information, being the raw seismic reflection-values. The black dot above represents the highest degree of information, where horizons, faults and other structures have been interpreted and

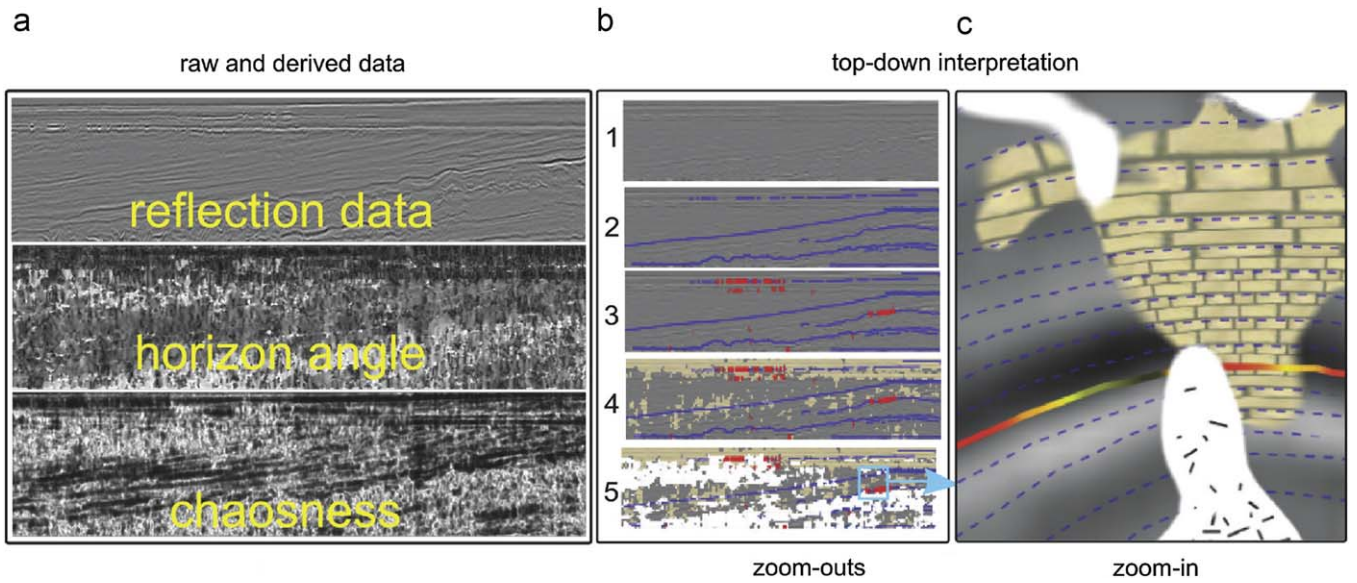


Fig. 3. Frame (a) shows the input data to be interpreted. Images 1–5 in frame (b) show the consecutive adding of illustrative layers on a highly zoomed-out slice as part of a top-down interpretation. Image 1 shows a zoom-out of the reflection data. Image 2 shows automatically generated lines that reveal the trends of the horizons. In image 3 areas of strong reflection values are shown with red lines. In image 4 areas of low-angled horizons are shown in brown corresponding to dark areas in the 'horizon angle' slice. Image 5 shows areas of badly defined horizons in white corresponding to high values in the 'chaosness' slice. Frame (c) shows a zoom-in on the indicated region of image 5.

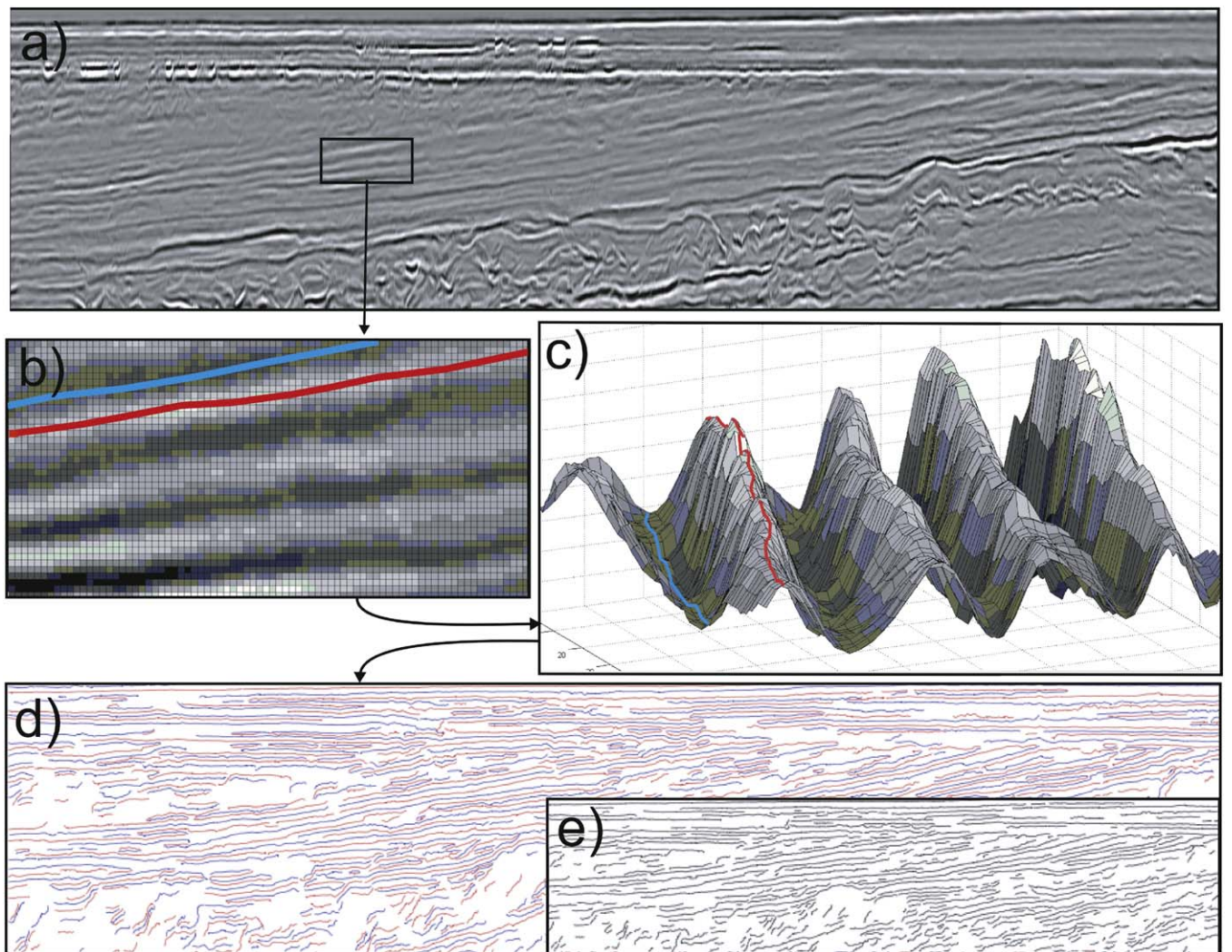


Fig. 4. Extracting horizons by tracing along peaks and valleys in the reflection data in (a). A peak is marked with red and a valley is marked in blue in (b) and (c). The rectangle in image (d) shows all extracted peaks and valleys from (a). Image (e) is a zoom-out of (d) with fewer horizons displayed to avoid crowding to demonstrate how an abstracted data representation enables sparse information visualization.

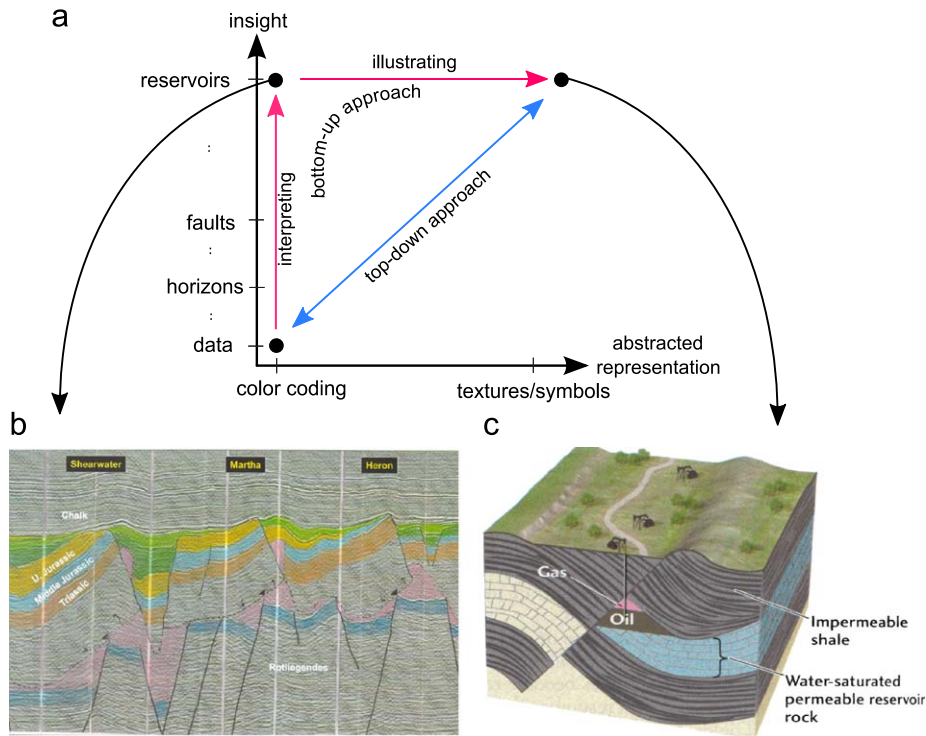


Fig. 5. Comparison of the bottom-up interpretation with our top-down approach. Image (b) is from Emery and Myers [8]. Image (c) is from Grotzinger et al. [10].

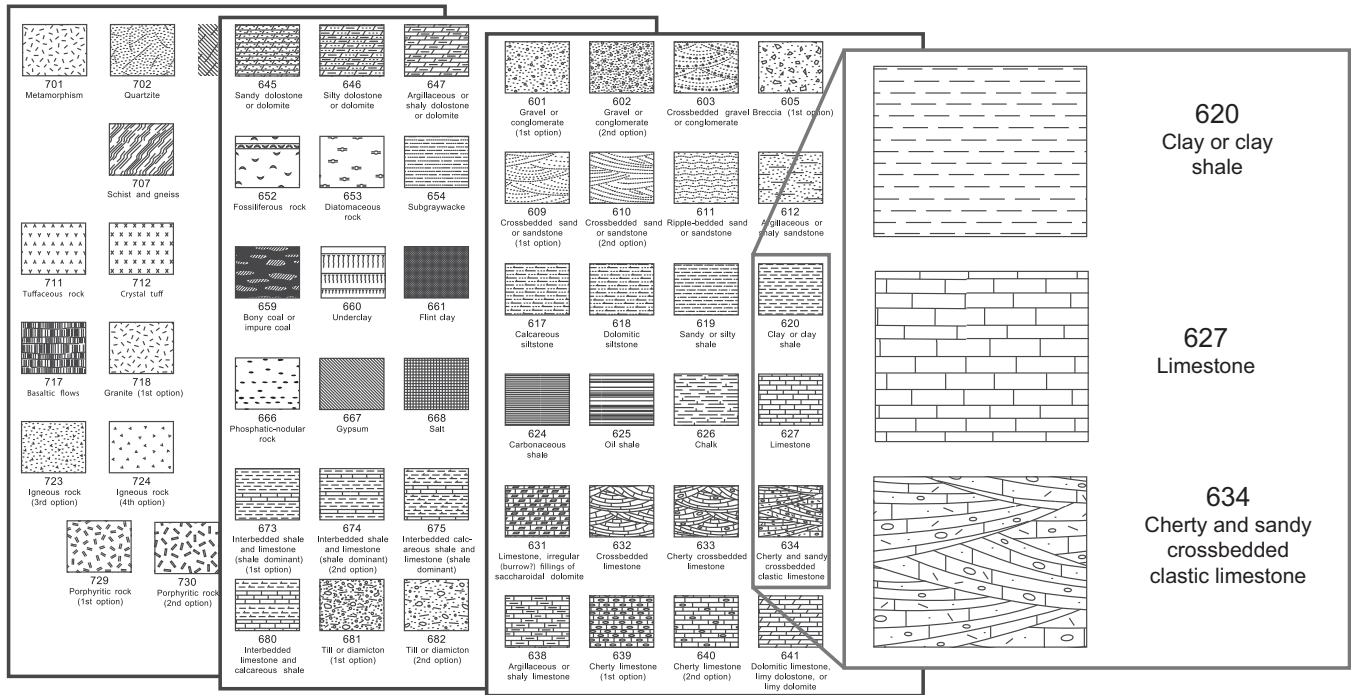


Fig. 6. Examples from a geological texture library [5] for annotating knowledge. Three of the textures are enlarged in the far right rectangle.

potential oil reservoirs have been found. An example of such an interpretation is seen in Fig. 5b. The horizontal axis in Fig. 5a indicates how abstracted information is represented. The further right, the more communicative the representation is, with textures and symbols vs color-coded data represented by the black dot close to the origin. An example of such a representation is given in Fig. 5c. A bottom-up interpretation is given by the vertical red line. It makes little or no use of illustrative techniques during interpretation. This is indicated by having the red vertical

line far to the left. After the interpretation, communication-friendly illustrations are generated. This is indicated by the red horizontal line at the top. Both interpreting and illustrating requires considerable manual work. In contrast, during a top-down interpretation, structures are found with computer assistance and the structures can be shown with automatic illustrative techniques. Computer-assisted interpretation enables vertical movements in the space of Fig. 5a and illustrative techniques enable horizontal movements. Creating a

communication-friendly interpretation is faster and has the advantage that the communicative content grows along with the increased knowledge. Thus the interpreter and the team can take advantage of the illustratively abstracted and communicative representation during analysis. This is in contrast to only having the representation available after both the interpretation is finished and the manual illustration has been made.

5. Representing knowledge with 2D textures

To ease communication, geologists and geoilustrators use a standardized language for representing knowledge. This language consists of textures for representing rock types and other information. The US Federal Geographic Data Committee (FGDC) has produced a document [5] with more than 100 standardized textures for rocks. These textures are referred to as lithological symbols in the geosciences. Fig. 6 shows three pages from this geological texture library. The oil company we have been collaborating with uses a colored version of the FGDC standard with some minor variations as shown in Fig. 7. Similarities of the two standards can be seen in the clay shale and the limestone textures in Figs. 6 and 7. Textures have the advantage that they can give an integrated visualization of layers, faults and attributes. The layer type and its orientation is communicated locally at any point on the texture. The way sediments have deposited and created the subsurface layers is very important to model during interpretation. For expressing such models, oriented textures are central. Finally, textures do not show sampling artifacts but are aesthetically pleasing even after extreme zooming, and visualizations can be more understandable by non-experts.

5.1. Computer-assisted annotation of knowledge

We create illustrative and knowledge-representing renderings by defining separate layers for each aspect of knowledge that is to

be communicated. The illustrative layers are then composited into one illustration. Each illustrative layer is created by a user defined mapping from raw data or processed data as seen in Fig. 3a to an abstracted representation. By mapping the data to the standardized representations used in the geosciences, the user creates a rendering that encodes seismic domain knowledge. The representations are either textures or lines that follow the trends of the underlying seismic data. Value ranges of the seismic data are mapped to different types of textures. Line styles have a user defined sparseness and transparency. See Patel et al. [18] for details on how the illustrative layers are specified.

To achieve the effect of textures and lines following the orientation trend of the underlying reflection data, we create a specific parameterization from the traced horizons. The parameterization ensures that the illustrative textures and straight lines are aligned with the extracted horizons. Illustrative layers in combination with sparseness control are key in performing a top-down interpretation. By using the appropriate sparseness and the appropriate number of layers, communicative illustrations can be made for any zoom level. An example of an overview picture using these techniques can be seen in Fig. 3. The raw data, i.e. reflection data, is given in the top slice of Fig. 3a. The two slices below are derived from the top slice. Other papers [11,18] give more information about derived attributes. A bottom-up interpretation would require that the user works on highly zoomed-in views of these slices and switches back and forth between them. Trying to get an overview of the reflection data by zooming out results in image 1 in Fig. 3b where few details are visible. However, by adding illustrative layers (images 2–5) with a sparse drawing style defined by the different modalities, it is possible to get a multiattribute overview of the data on a zoomed-out image. After an overview has been achieved, the interpreter can look closer at an area of interest (Fig. 3c).

In Fig. 7 another example of computer-assisted knowledge-annotation of seismic data is shown. The seismic data have been divided into rock layers and assigned textures representing rock

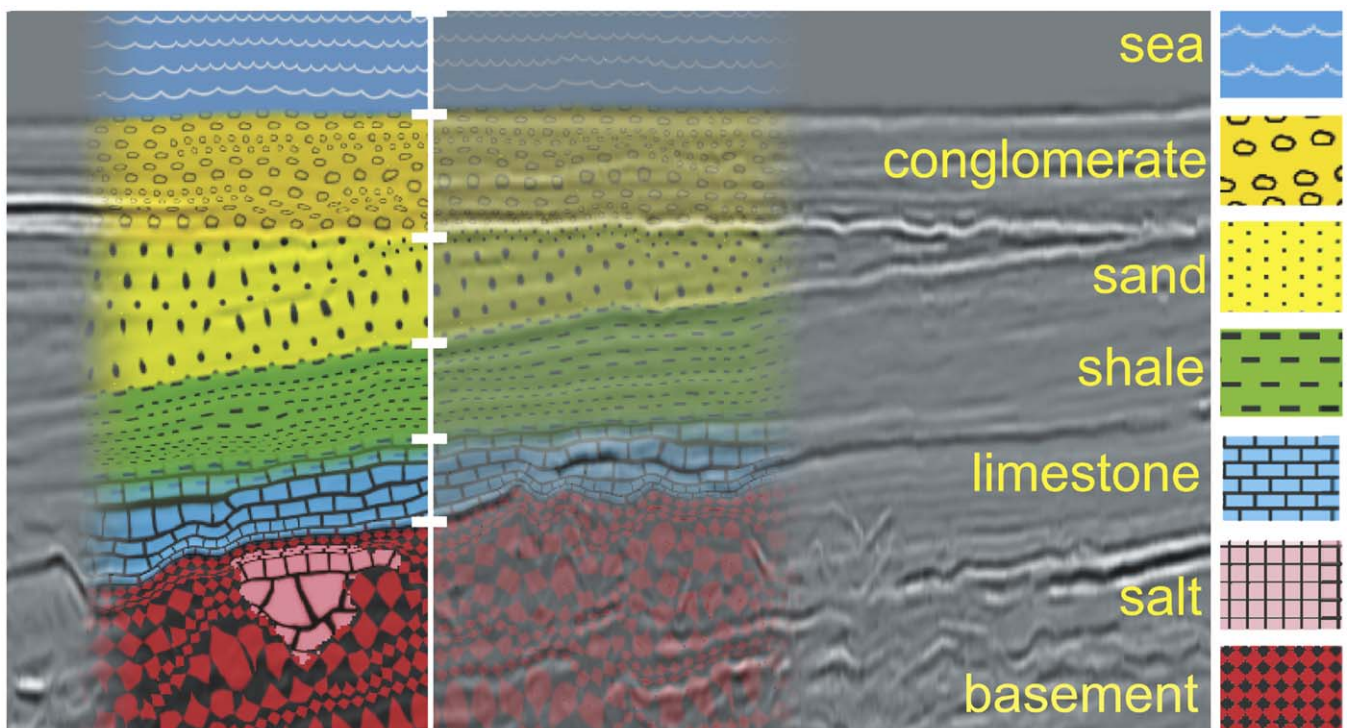


Fig. 7. A slice annotated by an interpreter. The legend to the right and the vertical white line in the middle has been laid over our original rendering. Low transparency is shown on the left side of the white line and high transparency is shown on the right side for demonstration purposes.

types. With computer-assisted annotation the illustrations can be created very fast. The interpreter only needs to assign different textures to depths along a vertical line through the slice (white line). Afterwards the textures are dragged out horizontally, to a user defined width, following the horizon trends. This is enabled by exploiting the calculated parameterization originally used for texturing. The salt area at the bottom is annotated by selecting precalculated horizon patches through mouse picking and assigning a specific texture to the horizon selection. The transparency of

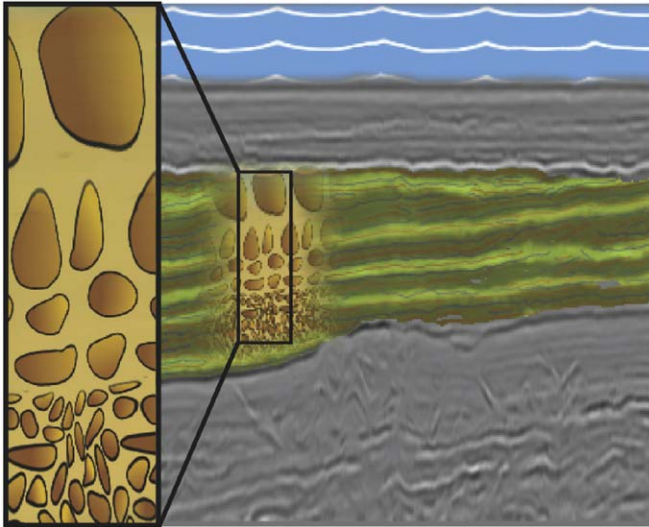


Fig. 8. Annotated seismic slice. Zoom-in on a texture showing sand grains with decreasing size along depth which represents a specific hypothesis.

the overlaid texturing can be changed so the underlying seismic can be seen in combination with the proposed rock subdivision for verification reasons. Two different transparencies are shown on the left and right side of the white line in Fig. 7. For more details, see Patel et al. [18].

A different example of computer-assisted annotation is shown in Fig. 8. The green area was annotated through mouse picking in the same way as the salt area in Fig. 7. The precalculated and selected horizon patches result in a green texture. The horizons themselves can be seen as faint red and blue lines corresponding to the peaks and valleys described in Fig. 4. The green textured area corresponds to the sand area in Fig. 7. The interpreter believed that this geologic layer was created in a sedimentation process and consequently formed a hypothesis that the sizes of the sand grains decrease with depth. To annotate this, different sand textures were assigned along the depth of the geologic layer and the textures were slightly dragged out horizontally. The sand textures are enlarged in the close-up view of Fig. 8.

With our techniques it is possible to smoothly move back and forth between visualization of data and visualization of knowledge as demonstrated in Fig. 9. The bottom image is a color-coded rendering of an impedance volume which has been calculated from the reflection volume. The impedance reveals the speed of sound through the rock. Blue denotes low impedance and green denotes high impedance. The top image shows the interpreted data with different textures for each rock layer. The middle image is a blend of the top and the bottom image where impedance can be seen in relation to the interpreted layers. On the right side of the middle image a correspondence between high impedance (green color) and its confinement inside one layer can be seen. Blendings make intermediate representations possible where both modes are displayed simultaneously. It enables the

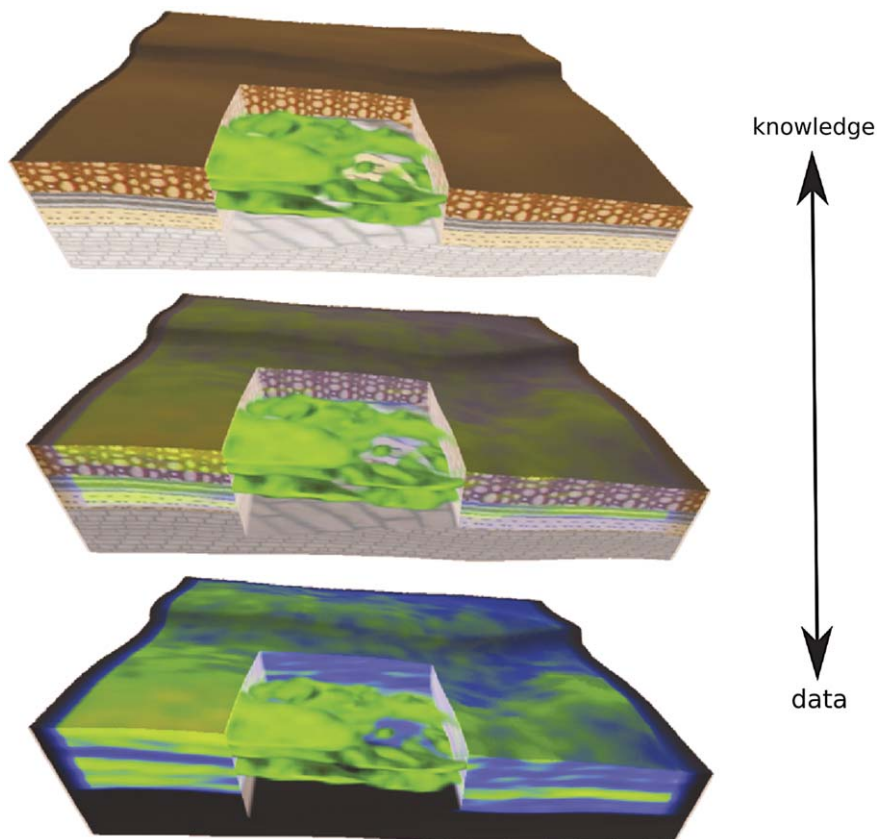


Fig. 9. The image series shows the transition from raw data visualization at the bottom to abstracted illustrative visualization at the top [19].

user to compare and verify the interpretation with the original data. More details can be found in the work of Patel et al. [19].

6. Representing knowledge with 3D textures

All standardized geologic textures are defined in 2D. However the Earth crust that is to be interpreted is inherently 3D. 2D textures do lend themselves directly to illustrate 2D seismic slices but have limitations when applied on 3D data as will be discussed in this section. On 3D data, 2D textures are frequently applied on planar cross sections. This technique is used in 3D geological illustrations. Several examples are given in Grotzinger et al. [10]. In this section we investigate using 3D textures for annotating 3D seismic data. Several advantages can be gained if suitable 3D textural representation of the 2D lithological symbols are specified and used. From an algorithmic point of view it is simpler to map 3D textures to 3D volumes and surfaces than 2D textures. From a perceptual point of view, 3D textures will reduce the problems of spatial and frame-to-frame incoherencies as will be discussed in the next

paragraph. Additionally 3D semitransparent textures may give rise to a higher perceptual depth. Volumetric variations can be revealed within the 3D texture and not only on the exterior boundary as is the case when using a 2D texture. Therefore in the context of knowledge-assisted visualization of 3D seismic data and as an outlook in the future, we explore 3D seismic textures.

6.1. 3D texture examples and comparison with 2D

Using 2D textures on 3D data as presented in this paper has several limitations. The dimensional mismatch between 2D textures and the 3D volume to apply textures on can lead to spatial incoherencies. 2D seismic textures are typically mapped to planar surfaces in 3D. Distortion problems arise with mappings to curved surfaces, and frame-to-frame incoherencies arise when interactively moving a textured cut-plane. In addition, using 2D textures on semitransparent volumetric regions is not well defined due to the dimensional mismatch. These problems can be solved by using appropriate 3D textures instead. However the

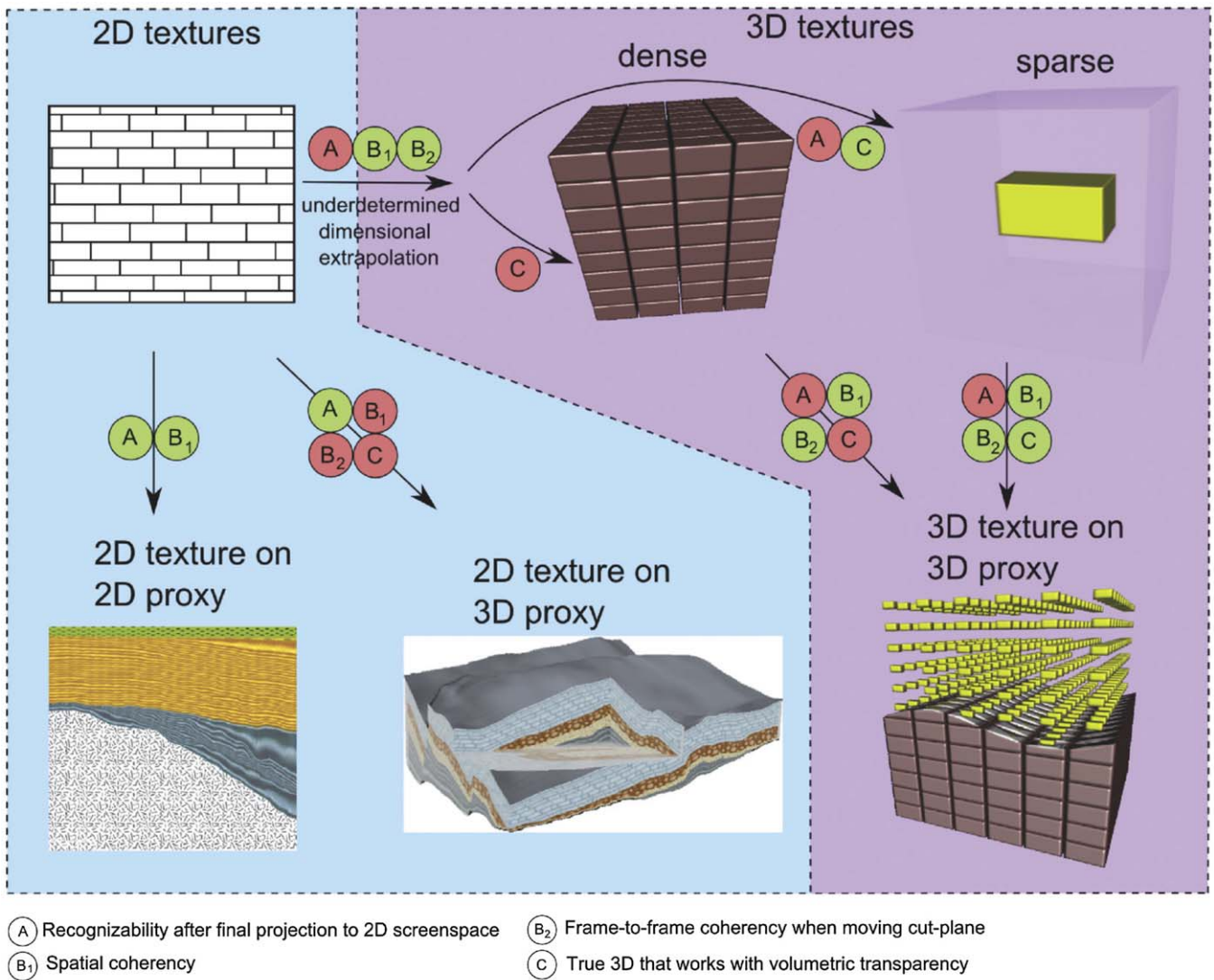


Fig. 10. An overview of advantages and disadvantages when using 2D vs 3D dense vs 3D sparse textures. The top row shows a 2D, a 3D dense and a 3D sparse texture. The arrows from left to right represent transformations of textures to 3D. The arrows from the top to the bottom row represent mappings of textures to geometries. The encircled characters on the arrows refer to perceptual advantages (green) and disadvantages (red) when performing these transformations and mappings. They are defined below the figure.

textures used for conveying knowledge in geosciences are only defined in 2D, there exist no 3D versions. If we want to use 3D textures, we must synthesize them ourselves from their 2D counterparts. Extending 2D textures to 3D has been attempted in other domains as discussed in the related-work section. However no work has been done on synthesizing 3D seismic textures and little work has been done on 3D textures with transparency in the context of scientific visualization. A dimensional extrapolation from a 2D texture to a 3D volumetric texture is underdetermined and has several possible mappings. We have looked at automatic algorithms for performing the extrapolation. They often have shortcomings when it comes to regular textures and in general when there is a specific mental expectation of how the 3D shape should look like that cannot be specified to the texture synthesizer. For instance creating a 3D texture from a 2D brick texture according to the algorithm of Kopf et al. [13] does not create a 3D cuboid brick texture as expected. Such algorithms take into account statistical properties and further constraints. They for example assume uniform distributions which are not always given for certain coherent structures.

Since the 2D–3D mapping is underdetermined it can result in dissimilarities between the rendering of a synthesized 3D texture and the rendering of the original 2D texture. Dissimilarities are unwanted as it is important that the viewer can relate a rendered 3D texture to its 2D origin. This is because the 2D texture has a connotation and represents knowledge. Thus there are respective advantages and disadvantages for either using 2D textures directly on surfaces in 3D or translating them to 3D and using them as volumetric textures.

We differentiate between dense 3D textures having no regions of full transparency, and sparse 3D textures having sparse opaque structures within transparent regions. We make this distinction due to their different perceptual characteristics. With dense 3D textures it is difficult to see the interior; however slices through these textures can be very similar to their 2D origin. With sparse 3D textures one can see the interior or can even see through which might reveal more information.

In Fig. 10 we list the advantages and disadvantages when applying 2D, 3D dense and 3D sparse textures on 2D and 3D regions. At the top of Fig. 10, examples of the three types of textures are given. The 3D dense and sparse textures will have other perceptual properties than the 2D exemplars they are synthesized from. Four perceptual characteristics, i.e., A, B₁, B₂ and C, are specified below the figure and listed on the transitional arrows between the texture types in the upper part of the figure. We refer to a 2D surface to be textured as a 2D proxy and to a 3D volumetric region to be textured as a 3D proxy. Textured 2D and 3D proxies are shown in the lower part of the figure. Different perceptual properties arise depending on the texture and proxy combinations. They are listed on the transitional arrows going from the top part with textures to the bottom part with proxies in Fig. 10. Gaining a property is annotated in green and losing a property is annotated in red. 2D textures applied on 2D proxies keep the textures' original appearance when disregarding rotational and perspective distortions, so the textures are recognizable (A). Problems with spatial coherency only show up on 3D textures thus property B₁ is present. The properties B₂ and C are not listed since they are undefined in 2D.

Applying 2D textures on 3D proxies in essence means applying 2D textures on 2D proxies positioned in 3D space. Therefore property A is also satisfied here. However the textures on the 2D proxies are not aware of their 3D embedding. This leads to inconsistent texturing on adjoining surfaces and on curved surfaces. Property B₁ is thus not satisfied for 2D textures on 3D proxies. The mapping of 2D textures is not synchronized with their 3D embedding. This leads to inconsistent texturing from

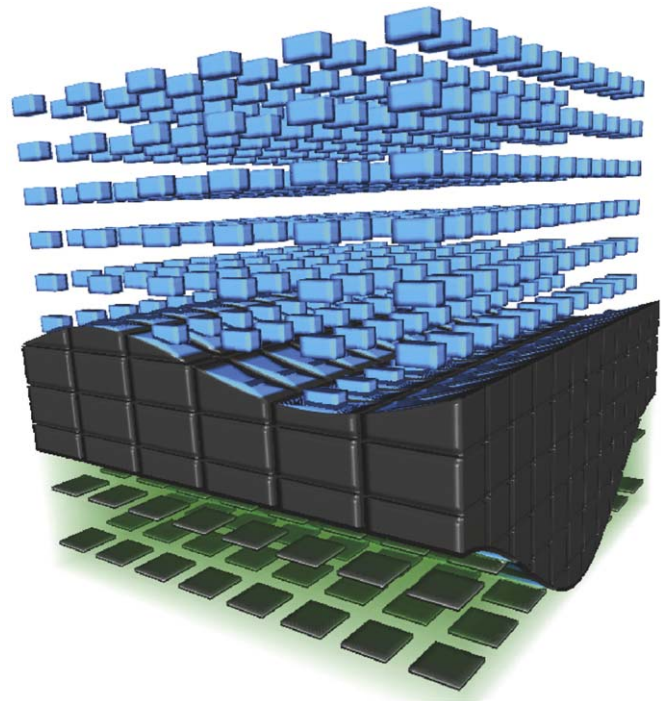


Fig. 11. An attempt at transforming the 2D limestone and shale seismic textures of Fig. 7 to 3D textures. The top layer is a sparse limestone 3D texture. The middle layer is a dense limestone 3D texture and the bottom texture is a sparse shale 3D texture.

frame-to-frame when interactively moving a 2D proxy such as a slice plane in 3D space. Thus property B₂ is not satisfied either. Finally, since the textures are not 3D they cannot be rendered in a meaningful way with volumetric transparency, therefore property C is also not satisfied.

In the following we discuss the transition arrow from 2D to 3D textures. The lack of spatial and frame-to-frame coherency and true 3D transparency can be resolved by creating 3D versions of the 2D textures. However the 3D textures can look different from their 2D originals depending on slice planes and projection angles. This means that property A is not satisfied. On the other hand, spatial coherency B₁ and frame-to-frame coherency B₂ is gained.

For a dense and opaque 3D texture property, C is not fulfilled. For a sparse 3D texture property, C is gained. Recognizing the original 2D texture might now be even more difficult than for a dense texture. The texture values accumulate in the depth direction during projection and clutter may arise due to the transparency. An example of this is shown in the top layer of Fig. 11.

Concerning the extrapolation of 2D textures to 3D dense textures, one option is to follow design guidelines to optimize the 3D texture for axis-aligned planar cuts. When restricting renderings to these cuts, the textures will have high recognizability. In this case property A is given at the expense of restrictions on how the volume can be sliced through.

The properties of using dense or sparse textures on a 3D proxy are shown on the respective arrows leading to the 3D proxy. Property A is negatively affected from both the extrapolation to 3D and from the sparse 3D texture representation.

In Fig. 11 we give an example of how the 2D shale and limestone textures defined in Fig. 7 may look like when transformed to 3D sparse and 3D dense textures. 3D textures might be better suited for communicating knowledge than 2D textures. With the example in Fig. 11 we briefly investigate this hypothesis. The top and middle layers in Fig. 11 represent

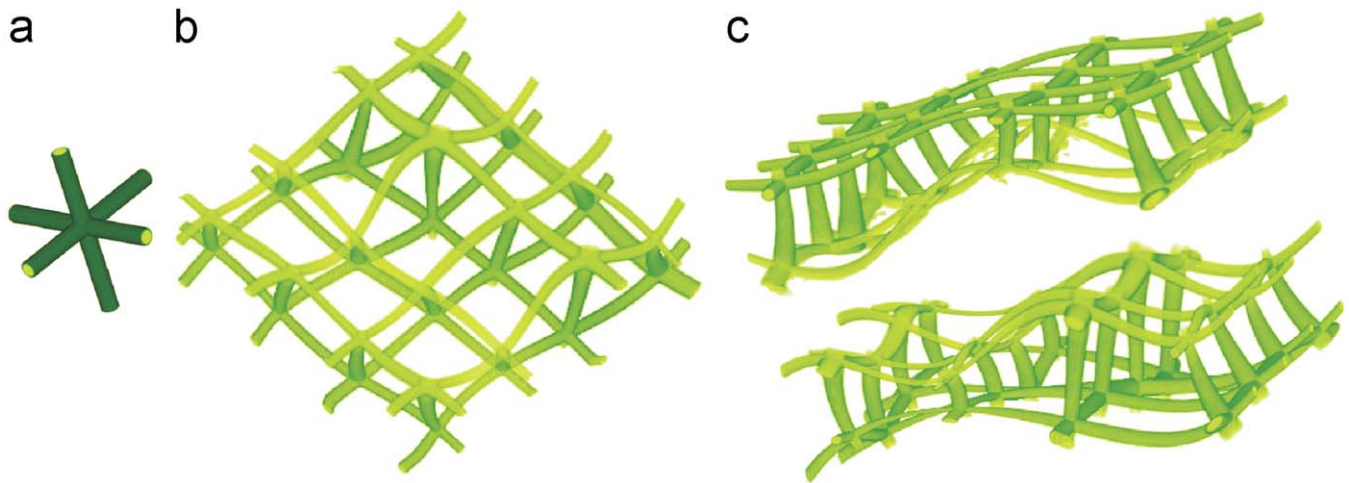


Fig. 12. (a) The texture primitive, (b) a deformed layer and (c) three deformed layers where the middle one is transparent.

limestone using a sparse and dense texture respectively. The dense limestone texture was designed with a blue core inside a black surrounding so that it will resemble the original 2D texture when sliced along axis aligned planes.

As an attempt at getting design guidelines for making the 3D textures we asked a geoscientific illustrator about the reasoning behind the design of the 2D seismic textures. The 2D textures stem from how the rock types look and break in nature and in borehole samples thus giving us concrete design guidelines for creating the 3D textures. Limestone will break in blocks due to its crystalline structure and shale is characterized by thin laminae. Therefore we extended the sparse limestone texture into blocks and the sparse shale texture was modeled with square sheets. We represented the green fill from the 2D shale texture as a highly transparent green haze and we made the square sheets black to preserve the colors from the 2D texture. With 3D textures the sparse top texture reveals the top surface of the middle layer. It allows insight into the data instead of only observing properties on the outer faces of the data as with 2D textures and 3D dense textures.

6.2. Seismic knowledge in deformed textures

Finally we discuss the effect of using deformed textures. Deformations carry knowledge as discussed in Section 5. If an oriented texture is used for a seismic layer, it communicates the orientation of the layer and the relative thickness of the layer through varying texture density. We have applied deformations on 3D sparse textures in Fig. 12. With a 3D sparse texture one can see the 3D deformation. Also the deformation throughout the whole layer is visible in one rendering. Obtaining this information with a 2D texture would require moving of a cut-plane through the volume. There are disadvantages of using 3D deformed textures as well. For example a sparse 3D texture does not show the deformation information with the same resolution as the 2D texture would. The problem of recognizability for deformed sparse textures is even more severe than for undeformed sparse textures.

7. Conclusions

We have presented the use of knowledge-assisted visualization through computer-assisted annotation for seismic interpretation. We have proposed to perform a top-down interpretation before the currently used bottom-up interpretation. This reduces the time for interpretation and for creating interactive communicative

illustrations. Standardized textures used for annotating seismic data were presented. Their applicability in knowledge-assisted visualization was shown and the positive implications were discussed. Furthermore we discussed the advantages and disadvantages of extending 2D seismic textures to 3D. The work presented here is still an ongoing research. A larger project including funding has been initiated by an oil company for integrating these ideas into the daily workflow of oil and gas interpretation.

References

- [1] Schlumberger information solutions (sis). Petrel, seismic interpretation software <<http://www.slb.com/content/services/software/support/petrelcorner/>>; 2007.
- [2] Bürger R, Hauser H. Visualization of multi-variate scientific data. In: EuroGraphics 2007 state of the art reports (STARS); 2007. p. 117–34.
- [3] Castanie L, Levy B, Bosquet F. Volumeexplorer: roaming large volumes to couple visualization and data processing for oil and gas exploration. In: Proceedings of IEEE visualization '05; 2005. p. 247–54.
- [4] Chen M, Ebert D, Hagen H, Laramée R, van Liere R, Ma K-L, et al. Data information and knowledge in visualization. In: IEEE visualization knowledge-assisted visualization workshop 2007.
- [5] Committee FGD, editor. Federal geographic data committee, Digital cartographic standard for geological map symbolization. FGDC-STD-013-2006 <www.fgdc.gov/standards/projects/FGDC-standards-projects/geo-symbol>, 2006.
- [6] Crawfis RA, Allison MJ. A scientific visualization synthesizer. In: VIS '91: proceedings of the 2nd conference on visualization '91, Los Alamitos, CA, USA: IEEE Computer Soc. Press; 1991. p. 262–7.
- [7] Dong F, Clapworthy G. Volumetric texture synthesis for non-photorealistic volume rendering of medical data. The Visual Computer 2005;21(7):463–73.
- [8] Emery D, Myers K, editors. Sequence stratigraphy. Oxford: Blackwell Scientific Publications; 2004.
- [9] Faraklioti M, Petrou M. Horizon picking in 3D seismic data volumes. Machine Vision and Applications 2004;15(4):216–19.
- [10] Grotzinger J, Jordan TH, Press F, Siever R. Understanding Earth. New York: Freeman; 1994.
- [11] Iske A, Randen T, editors. Atlas of 3D seismic attributes, mathematics in industry, mathematical methods and modelling in hydrocarbon exploration and production. Berlin, Heidelberg: Springer; 2006.
- [12] Kirby RM, Marmanis H, Laidlaw DH. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In: Ebert D, Gross M, Hamann B, editors, IEEE visualization '99. San Francisco; 1999. p. 333–40.
- [13] Kopf J, Fu C-W, Cohen-Or D, Deussen O, Lischinski D, Wong T-T. Solid texture synthesis from 2D exemplars. In: ACM transactions on graphics (Proceedings of SIGGRAPH 2007), 2007. p. 2:1–9.
- [14] Lidal EM, Langeland T, Giertsen C, Grimsgaard J, Helland R. A decade of increased oil recovery in virtual reality. IEEE Computer Graphics and Applications 2007;27(6):94–7.
- [15] Lu A, Ebert DS. Example-based volume illustrations. In: Proceedings of IEEE visualization 2005, 2005. p. 83–92.
- [16] Lu A, Ebert DS, Qiao W, Kraus M, Mora B. Volume illustration using wang cubes. ACM Transactions on Graphics (TOG) 2007;26(2).

- [17] Owada S, Nielsen F, Okabe M, Igarashi T. Volumetric illustration: designing 3D models with internal textures. In: Proceedings of the 2004 SIGGRAPH conference, 2004. p. 322–8.
- [18] Patel D, Giertsen C, Thurmond J, Gjelberg J, Gröller ME. The seismic analyzer—interpreting and illustrating 2D seismic data. *IEEE Transaction on Visualization and Computer Graphics* 2008;14(6):1571–8.
- [19] Patel D, Giertsen C, Thurmond J, Gröller ME. Illustrative rendering of seismic data. In: Lensch HPA, Rosenhahn B, editors. *Proceeding of vision modeling and visualization 2007*; November 2007. p. 13–22.
- [20] Pepper R, Bejarano G. Advances in seismic fault interpretation automation. In: *Search and Discovery article 40170*, Poster presentation at AAPG annual convention, AA, 2005. p. 19–22.
- [21] Plate J, Tirtasana M, Carmona R, Fröhlich B. Octreemizer: a hierarchical approach for interactive roaming through very large volumes. In: *Proceedings of VISSYM '02*; 2002. p. 53–64.
- [22] Rautek P, Bruckner S, Gröller E. Illustrative visualization—new technology or useless tautology? *Computer Graphics Quarterly, VisFiles* 2008; 42(3).
- [23] Ropinski T, Steinicke F, Hinrichs KH. Visual exploration of seismic volume datasets. In: *Journal proceedings of WSCG '06*, vol. 14, 2006. p. 73–80.
- [24] Taylor RM. Visualizing multiple fields on the same surface. *IEEE Computer Graphics and Applications* 2002;22(3):6–10.
- [25] Viola I, Sousa MC, Ebert D, Andrews B, Gooch B, Bruckner S, et al. Illustrative visualization for science and medicine full-day tutorial. In: *IEEE visualization, 2006*.
- [26] Wang L, Mueller K. Generating subresolution detail in images and volumes using constrained texture synthesis. In: *Proceedings of IEEE visualization 2004*; 2004. p. 75–82.