

Seismic Volume Visualization for Horizon Extraction

Daniel Patel*
Christian Michelsen
Research, Bergen,
Norway

Stefan Bruckner†
Institute of Computer Graphics
and Algorithms, Vienna
University of Technology, Austria

Ivan Viola ‡
Department of
Informatics, University of
Bergen, Norway

Eduard M. Gröller §
Institute of Computer Graphics
and Algorithms, Vienna
University of Technology, Austria

ABSTRACT

Seismic horizons indicate change in rock properties and are central in geoscience interpretation. Traditional interpretation systems involve time consuming and repetitive manual volumetric seeding for horizon growing. We present a novel system for rapidly interpreting and visualizing seismic volumetric data. First we extract horizon surface-parts by preprocessing the seismic data. Then during interaction the user can assemble in realtime the horizon parts into horizons. Traditional interpretation systems use gradient-based illumination models in the rendering of the seismic volume and polygon rendering of horizon surfaces. We employ realtime gradient-free forward-scattering in the rendering of seismic volumes yielding results similar to high-quality global illumination. We use an implicit surface representation of horizons allowing for a seamless integration of horizon rendering and volume rendering. We present a collection of novel techniques constituting an interpretation and visualization system highly tailored to seismic data interpretation.

Keywords: Seismic interpretation, Seismic horizons, Volume rendering, Ambient occlusion, GPU acceleration.

1 INTRODUCTION

Energy is one of the most important resources in today's societies. Over half of the world-energy needs are covered by oil and gas [13]. This results in high demands for hydrocarbon resources and makes their identification and extraction economically valuable. To identify hydrocarbon reservoirs, subsurface volumetric data is obtained by sending sound waves into the ground and receiving and processing their echoes. The resulting seismic reflection volume then goes through a complex and time consuming manual interpretation for identifying subsurface structures which may hint to where hydrocarbons are trapped. In this paper we present a system for rapid interpretation of seismic reflection volumes.

The earth subsurface consists of material layers with distinct mineral densities and porosity characteristics. The interfaces between material layers are called horizons. They are indicated as high and low valued bands in seismic reflection data and are central structures for interpretation. Other structures such as faults, channels, salt bodies, and gas chimneys are mainly identifiable by their interaction with horizons. Faults are generally sub-vertical fractures which have shifted horizons on either sides; they are thus identified as discontinuities in horizons. Salt bodies are homogeneous units of salt. Due to the high seismic wave velocity of homogeneous salt, such structures can have strong reflections at their boundaries and low or no internal reflections. Areas void of horizons can thus indicate the presence of salt bodies having the property of shadowing the underlying seismic data. Subsurface leakages of gas, called gas chimneys, can be indicated by the up-bulging of horizons around

them and the fragmentation of the horizons in their path. A reservoir in a porous rock formation can be accurately pinpointed by identifying its upper and lower boundary. The different rock materials at the boundaries give rise to horizons in the reflection data. Therefore horizons can be central in delineating reservoirs.

In addition to these descriptive properties of horizons, they are perhaps the most easily identifiable objects in the seismic volume and thus among the most important objects during interpretation. Compared to most other seismic structures, horizons can directly be identified with image processing techniques such as ridge and valley detection. For these reasons we focus on horizon identification in this work. We present an approach for automatically identifying horizons and visualizing them.

Visualizing 3D seismic data is challenging due to the dense nature of the data, the high presence of noise, and the difficulty to identify, manipulate and select structures in 3D. Consequently, interpretation of 3D seismic data is nowadays split into lengthy 2D interpretations of a stack of selected cross-sections throughout the 3D volume. In this paper we present a novel approach that facilitates rapid 3D interpretation using interactive computer-assisted 3D visualization. The basic idea is to precompute horizon candidates from the dataset and partition them into small building blocks. The building blocks are then selected by the geoscientist during a rapid interpretation process and assembled together into correct horizons.

As opposed to the accurate but slow interpretation process currently used, ours is quick but less accurate. However the interpreter is aware of the reduced accuracy since the seismic ground-truth data is provided for context. Our approach creates early overviews for identifying areas to later perform accurate interpretation in, instead of performing accurate and time consuming interpretation of all the data. The currently used interpretation workflows do not support attaining overviews. We propose rapid sketching of 3D interpretations. This is similar to 3D sketch systems [10] in architectural CAD systems which have become superior during idea prototyping over traditional 2D floor-based modeling software [2]. For subsequent detailed modifications, interpreted horizons can be imported into existing interpretation systems. In such detail-oriented systems, tasks like identifying horizon-splitting structures such as faults and unconformities can be performed.

A major challenge for providing useful 3D interactive visualization, is the choice of an appropriate 3D rendering algorithm. Gradient-based shading [17] is effective for depicting volumetric data having clear material boundaries. The gradient vectors are then used as normals in the Phong illumination model [24]. Gradient-based methods, whether based on central differences or more complex seismic dip and azimuth estimations [19], are in general sensitive to high-frequency noise. As seismic acquisition is based on acoustics, data are typically of noisy nature and lack distinct material boundaries. Gradient-based shading on seismic data introduces distracting artifacts which makes interpreting 3D seismic data renderings difficult. Other approaches, such as unshaded direct volume rendering tend to depict seismic data as a homogeneous cloud without distinct features [33]. Common approaches are frequently unsuitable for visualizing seismic data. In this paper we present a gradient-free ambient-occlusion like method for shaded volume rendering of seismic data that reduces the abovementioned problems.

*e-mail:daniel@cmr.no

†e-mail:bruckner@cg.tuwien.ac.at

‡e-mail:ivan.viola@uib.no

§e-mail:groeller@cg.tuwien.ac.at

Related work is reviewed in Section 2. The high-level concept is described in Section 3. Sections 4 and 5 describe the horizon extraction and visualization stage. A demonstration of the proposed technology is presented in Section 6 and conclusions are drawn in Section 7.

2 RELATED WORK

Several research works and commercial solutions present interpretation and visualization algorithms for 3D seismic data. Pepper and Bejarano [23] gives an overview of computer-assisted interpretation algorithms. Commercial software used in oil companies include HydroVR [18] and Petrel [30]. Interpretation algorithms for horizons are typically semi-automatic and require a detailed and time consuming user involvement. User steered horizon growing is a standard method. The user manually places a seed point on a horizon in a seismic slice and adjusts growing parameters before starting a growing process of the horizon. This method is not fully interactive due to the need to set parameters and to wait for the growing to finish. To our knowledge there are no commercial solutions that support fully interactive and thereby rapid horizon interpretation of seismic data. For generality, the interpretation software often presents the user with many parameters to specify for the growing process. Typically the user changes parameters and re-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 for a geologist performing interpretation. One might argue that too much low level control is given to the user. In our system we aim at minimizing the need for parameter tweaking.

Growing algorithms based on local waveform matching are common. The local waveform is defined as the vertical 1D signal of the seismic values in a neighborhood above and below a sample. Castanie et al. [7] propose user-specified seeding followed by growing in areas that have local waveforms similar to the seedpoint. The method builds on the fact that horizons typically have characteristic local waveforms. The method requires setting parameters such as the length of the neighborhood to match. Interpretation software [30] performs growing in voxels that have been thresholded or in zero crossings or extrema of the local waveform. We employ the latter method as it requires no parameters to be set.

There are completely automatic approaches for horizon interpretation [4]. In their work, voxels are mapped to points in an n-dimensional space based on their local waveforms. Clusters are then identified in the n-dimensional point cloud. Probably due to the low control and low predictability of the outcome and long cycles of setting parameters and waiting periods before the results are available, such methods have not gained popularity.

Automatic horizon interpretation as preprocessing before rapid interpretation of 2D seismic slices has been proposed by Patel et al. [21]. An extension of their technique into 3D would be difficult as their horizon analysis produces just a set of line segments for each 2D seismic slice. A 3D horizon preprocessing method was presented in the work by Farakliti and Petrou [9]. However their connected component analysis was only able to grow planar and well-defined horizons.

In our work we present the concept of rapid horizon interpretation by focusing on the analysis of seismic horizons, high quality 3D visualization and quick interaction techniques. We automatically extract surfaces that, with high likelihood, coincide with horizons. We subdivide these surfaces into smaller surfaces using an existing mesh clustering algorithm [1]. An overview of mesh clustering algorithms can be found in the work by Shamir [32]. Shamir categorizes clustering methods into greedy quick methods, global slow methods, growing methods, hierarchical methods and spectral analysis methods. We chose a hierarchical greedy method due to the generation of hierarchy information which we use during interpretation. As opposed to growing methods the approach does not

require initial seed faces as input that affect the resulting subdivision and is therefore deterministic.

For a seamless integration of horizon-surface visualization with seismic-volume visualization, we represent the surfaces as a distance volume having segmentation masks around each surface-part. Existing distance-transform techniques have been surveyed by Jones et al. [15]. Since we perform the distance transform in a preprocessing step, we are not dependent on speed. We create a computationally expensive but analytically accurate distance transform by in essence considering the analytical distance to all triangles from each voxel [14].

The basic concept of our approach is to carry out horizon interpretation directly in 3D. Several aspects for 3D visualization of seismic volumes have been investigated in earlier works. Plate et al. [26, 25] and Castanie et al. [7] discuss handling multiple large seismic volumes in the rendering pipeline. Ropinski et al. [29] discuss volume rendering of seismic data in VR with volumetric cutouts. Illustrative rendering techniques employing textures have been proposed for presentation of interpreted seismic data [22].

Often gradient-based illumination is used to render volumes and calculate the gradient directly from the seismic data. One exception is the work by Silva et al. [33]. They observe that seismic horizons are not isosurfaces of the volumetric data, therefore seismic gradients are not optimal to use. Instead they calculate gradients from a derived phase volume which gives better results. However, gradient-based illumination models lack depth cues and display a strong visual dominance of noise making it difficult to identify subtle horizon structures. For these reasons the rendering approaches currently used for seismic data are not ideal for horizon display.

Depth cues can be added to volume rendered images by employing more realistic illumination models. Yagel et al. [35] employ recursive ray tracing which allows effects such as specular reflection and shadows. Behrens and Ratering [3] add shadows to texture-based volume rendering. The model presented by Kniss et al. [16] captures volumetric light attenuation effects including volumetric shadows, phase functions, forward scattering, and chromatic attenuation. Rezk-Salama [27] presents an approach for GPU-based Monte Carlo raytracing. Max [20] gives a comprehensive overview of different optical models for volume rendering. A problem of increasing the physical realism is, however, lack of speed and lack of control over the specific appearance of certain structures of interest. As they are based on actual physical laws, it is difficult to control individual visualization properties separately.

Other approaches employ quicker, visually plausible approximations of realistic lighting effects. Stewart [34] introduces vicinity shading, a view-independent model to enhance perception of volume data. It is based on occlusions in the local vicinity of a sample point resulting in shadows in depressions and crevices. Similarly, Hernell et al. [12] use a local approximation of ambient occlusion. Real-time global illumination for volumes has been done by Hernell et al. [11] by local piecewise integration and subsampling using adaptive compression. Desgranges et al. [8] use incremental blurring to achieve shading effects without the use of a gradient. The approach by Bruckner and Gröller [6] is able to generate shadowing and emission effects in a view-dependent manner while still allowing interactive transfer-function modifications. We use an interactive gradient-free illumination model inspired by Schott et al. [31] to visually reduce the noise and to provide depth cues.

3 RAPID HORIZON-INTERPRETATION PIPELINE

An overview of the pipeline of this paper is depicted in Figure 1. The pipeline is divided into two parts. The first part is a preprocessing step that extracts horizon candidates in advance of the visualization so the user can perform rapid interpretation. This is covered in Section 4. The second part, the realtime use of the system, is discussed in Section 5 and covers visualization of volumes (5.1) and horizons (5.2) and user interaction for interpretation (5.3).

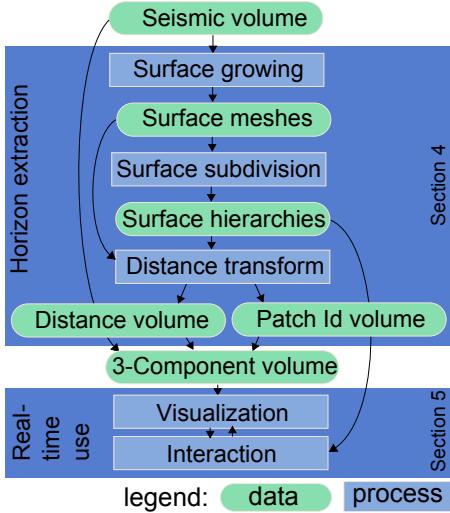


Figure 1: An overview of our pipeline. All preprocessing is performed in the horizon-extraction phase, followed by visualization and interaction for assembling correct horizons. The horizon extraction is further described in Figure 2 and Figure 4.

4 HORIZON EXTRACTION AND REPRESENTATION

Performing automatic feature extraction on seismic data is difficult due to the degree of uncertainty and noise in seismic data. Even domain experts may come up with different interpretations which demonstrates the under-defined nature of seismic data.

We design our horizon extraction with this uncertainty in mind by providing the user with a large collection of precalculated horizon surfaces. A set of plausible horizon surface-parts based on a horizon-growing and a surface-subdivision algorithm is precalculated and presented to the user to pick from. The user can then build horizons by assembling these surface-parts. The user is now freed from being repeatedly interrupted during interpretation by having to explicitly seed horizons followed by waiting for the horizons to grow. Explicit methods have advantages when it comes to fine-tune the horizons. However we focus on rapid interpretation for getting a first overview of the data. This overview can complement existing workflows. Later, a detailed interpretation can be focused on areas that have been identified as important with our method.

We enable the user to choose which level of horizon granularity she wishes to operate on. For each picked horizon surface she can change the granularity level. For this to work we first grow all possible horizons in the volume. Then we perform a hierarchical subdivision of the horizons into smaller surfaces. The hierarchy is stored in a tree structure with the smallest surfaces as leaf nodes. The user can navigate in the hierarchy and work with surfaces of the size and shape which reflect the interpretation best.

Horizon identification methods that do not rely on growing, such as performing clustering of the voxels based on their local neighborhood [4], result in segmentation masks that are not directly transferable to polygonal surfaces. Incremental growing methods are better suited as they can be made topology aware. This allows for constrained growing and for producing connectivity information during the polygonization of the horizons. We seed and grow each voxel in the volume. Growing is only performed in voxels that are on positive extrema of the vertical waveform, i.e. voxels that have a lower valued voxel directly above and below it. This gives a robust tracing of horizons of varying intensities. However other growing criteria can easily be used instead. We achieve subvoxel precision on the horizon growing by fitting a higher order polynomial to the trace signal and using the exact position of the analytical maximum.

This avoids staircase artifacts aligned with the voxels.

Voxels are visited only once to avoid intersecting and overlapping surfaces. An example of growing from one voxel is shown in Figure 2. The width of each growing frontier, defined as the number of neighboring points it consists of, must be above a certain size or else the growing of that border is terminated. The larger the minimum width is, the less surface bifurcations are created. For instance, a minimum width of 1 will create surfaces that branch out in lines which in turn may become surfaces. The width parameter heuristically controls the minimum radius a surface can have. We achieved a good balance between undetected horizons and over-segmentation by using a width of four voxels. To create surfaces, we triangulate the frontier during growing. We achieve this by triangulating the strips consisting of the points in the current borders (green points in Figure 2) and the points in the borders from the previous step (red points in Figure 2). The grown surface is represented as a vertex list of the coordinates of the grown voxels followed by an index list of triangle triples with indices to these vertices.

Frequently during surface growing, parts which an interpreter would consider as separate horizons, are grown into a single surface. This typically happens for horizons vertically close to each other. Due to noise or unconformities, a growing frontier might jump to the horizon directly above or below. It will then start growing on that horizon while other frontiers continue on the original horizon. Erroneous growings result in surfaces consisting of multiple merged horizons instead of isolated correct ones. An example of such an incorrect growing originating from one seed point and resulting in five overlapping horizons can be seen in Figure 3. To address overgrowing, we subdivide each grown surface into smaller pieces for the user to select. The user can in realtime pick such surfaces, subdivide them further if necessary and thereby assemble correct horizon surfaces.

For subdividing the triangulated surfaces into smaller pieces we are using the method described by Attene et al. [1]. This greedy method creates a hierarchical surface segmentation by starting with single-triangle surfaces and iteratively merging neighboring surfaces into larger ones by choosing the neighbors that maximize the flatness of the joined surface. This process continues until all surfaces are merged into one. See Figure 3 for an illustration of the subdivision process. The merging operations can be described as the interior nodes of a binary tree having single triangles as leaf nodes. We store this binary surface tree for each grown surface. The binary tree is used later during interpretation. It lets the user work with subsurfaces at a user-defined granularity level by moving up and down in the hierarchy. After selecting a leaf node, the user can navigate up in the tree and select surfaces of appropriate sizes for a quick horizon assembly.

The subdivisions of a binary surface tree are depicted in Figure 3. For each consecutive image, a jump of four levels up in the tree is performed. Distinct colors represent distinct nodes in the corresponding tree level. In the top image, the tree leaves are seen. Each color represents a single triangle of the surface. However, single triangles have approximately the size of a voxel and are prohibitively small to work with. Therefore we prune the tree so that during interaction, the child nodes are not single triangles, but groups of triangles of a predefined minimum size. We chose to prune the tree into having leaf nodes with a minimum of 500 triangles. Each leaf node consists of a unique leaf-id followed by a list of indices to the triangle triples of the surface as described earlier. Each interior node consists of pointers to two child nodes and the number of triangles in its subtree.

For an integrated representation of seismic data and surface trees, we perform a distance transform on all leaf surfaces which together constitute all grown surfaces. In each voxel of the seismic volume we calculate and store the distance to the closest point on a leaf surface together with the unique id of that leaf surface. Then, during picking, the picked voxel's id is a link to the leaf node in the surface

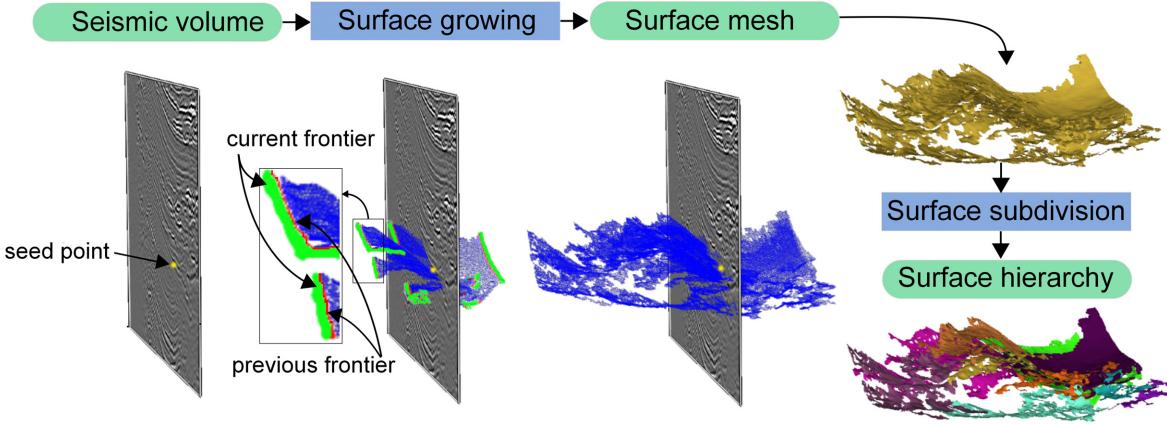


Figure 2: The process of growing from one specific seedpoint (yellow point). The green borders show the growing frontier and the red regions show the previous frontier of the blue triangulation. The triangulation is then subdivided.

tree of the closest surface. This link enables 3D picking of horizons followed by a hierarchical navigation in the surface tree from the leaf node to the largest valid horizon surface. We combine the seismic volume, the distance field volume and the leaf surface-id into one 3-component volume. This facilitates fast lookups during interaction. See Figure 4 bottom for an illustration of the distance component on a slice. The distance component combined with the leaf surface-id component allows for several advanced volume-rendering modes elaborated in Section 5.2.

Hierarchies with leaf surface-ids and geometric surface definitions associated with each leaf surface-id are stored on file. After the user has built horizons by selecting leaf surface-ids, the interpreted horizons can be geometrically reassembled and exported into commercial seismic interpretation systems for further processing.

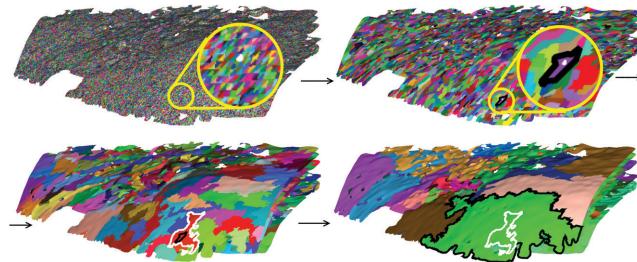


Figure 3: Increasing levels of a surface hierarchy made from one grown surface. The top-left image shows the initial one-triangle surfaces. Each successive image shows the resulting merging after four merge levels in the binary hierarchy tree. One triangle is magnified in white at the top image and its successive higher level surfaces are shown delineated with alternating white and black borders.

5 VOLUME AND HORIZON VISUALIZATION WITH INTERACTION

During interactive interpretation, high quality volumetric rendering is performed, horizons are visualized and the user can interact with the data. These three topics are covered respectively in the three following subsections.

5.1 Gradient-Free Volume Visualization

Conventional surface-based shading using the gradient direction as a substitute for the surface normal is most effective when there

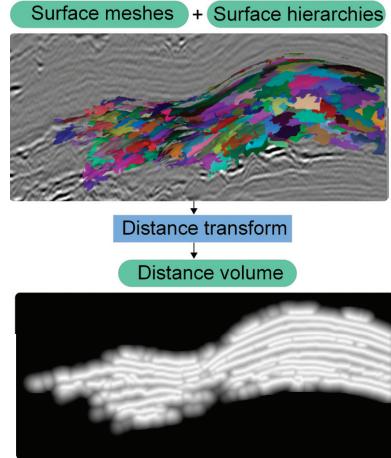


Figure 4: The distance transform. One grown and subdivided surface mesh is shown in 3D at the top intersecting a slice of the seismic data. The colors represent surfaces at the lowest level in the pruned hierarchy. The bottom image shows a slice through the resulting distance transform with white representing distance zero and darker colors representing higher distances.

are clearly identifiable material boundaries. An example is medical volume data acquired by computed tomography (CT). For seismic data, however, this approach commonly results in noisy images which are difficult to interpret. Unshaded visualization, on the other hand, suffers from a lack of clearly delineated features. In our approach we want to identify horizons and then visualize the extracted horizon information in conjunction with the original seismic data to enable verification of the extraction process. Thus, a volume visualization technique is required which gives a good indication of the overall structure of the data set.

A candidate to fulfill these requirements is ambient occlusion. The incoming radiance at a point is determined by shooting rays in all directions from the point for finding its degree of occlusion by other parts of the volume. The result is an approximation of global diffuse illumination. It produces soft shadowing effects which give a good indication of spatial relationships. In volume visualization the opacity at any point is determined by the transfer function. Ambient occlusion therefore requires an expensive computation step every time the transfer function is modified. The

approach of Ropinski et al. [28] relies on local histogram clustering to precompute this information for all possible transfer function settings. The precomputation is extremely time and memory consuming. In addition, the high frequency and noisy seismic data makes a bad candidate for local histogram clustering. We therefore choose to employ a view-dependent method which generates similar shadowing effects but still allows interactive transfer function manipulation.

Inspired by the approach of Bruckner et al. [6], Desgranges et al. [8] as well as recent work by Schott et al. [31], we use a slice-based volume rendering method where an incremental blurring operation is applied to the accumulated opacity buffer. This buffer is then used to determine the degree of shadowing applied to subsequent slices. The volume is traversed in front-to-back order using view-aligned slicing. Two buffers C_i , the color buffer, and O_i , the occlusion buffer, are written for each slice i . For each sample on the slice, we apply a Gaussian low-pass filter G to the occlusion buffer O_{i-1} of the previous slice and combine it with the opacity contribution α_i of the current slice using additive blending:

$$O_i = G * O_{i-1} + \alpha_i$$

Based on this incremental filter operation, for each sample on the slice, a modulation factor λ_i is computed which determines the degree of visibility:

$$\lambda_i = \frac{1}{1 + G * O_{i-1}}$$

The color contribution c_i is then multiplied by this factor and c_i and α_i are used for conventional alpha-blending using the over operator to generate the color buffer C_i .

This simple approach generates soft shadowing effects akin to ambient occlusion but can be performed interactively on current graphics hardware. Due to the incremental filtering of the occlusion buffer, darkening from a highly occlusive region will reduce smoothly with distance leading to a visual appearance as if the volume was illuminated by a large area light source located at the camera position. By setting ambient values in the occlusion buffer for certain objects in the volume one achieves the effect of having objects cast light instead of shadows. Examples of such light emitting objects will be shown in the subsequent section.

In Figure 5 we give a comparison between a Phong-shaded seismic volume and our rendering approach. As can be seen, our rendering is less susceptible to high frequency noise and gives a better depth perception of the data. Both images use the same transfer function for comparison reasons with a simple black to white color ramp. The lower values of the reflection data are set to transparent while the higher values are set to be opaque.

5.2 Horizon Visualization

Horizons are represented implicitly by the distance component and the leaf surface-id component. As opposed to a geometric model, this representation enables a seamless co-rendering of horizons and seismic data in one rendering pass. Implicitly representing the surfaces removes the need to store and update visualization parameters of the surface geometries in memory. It enables a single-pass integrated rendering, however with the penalty of an increased volume size. The distance volume is of the same resolution as the seismic volume. Due to trilinear interpolation, the implicit surfaces will therefore be smooth with an error distance from the original polygon of less than one voxel.

Having a volumetric segmentation mask with distance information for horizons opens up the possibility for more advanced rendering techniques. The picked leaf surface-ids and the distance volume constitute a segmentation mask around a horizon. This segmentation mask can be used to render horizon surfaces with a user defined appearance and thickness. A mapping from leaf surface-ids

to RGBA and emission values defines the color, opacity and emission of each leaf-surface in the volume. During the slice-based volume rendering, the distance and leaf surface-id component of the pixels of each slice that is composited, is looked up in the volume. If the distance is less than the user defined horizon thickness, the pixel is modulated with the color and transparency assigned to the leaf surface-id. Initially all horizon leaf surface-ids are set to fully transparent. Thus the mapping defines the selection and appearance of horizons. Interpreted horizons are then easily visualized with different transparencies, colors and emission values.

Volume rendering of the seismic data only, can be restricted to take place in the vicinity of selected horizons and having full transparency everywhere else for verification reasons. Alternatively in an inverse mode, full transparency can be applied in the segmentation mask. This removes interpreted structures from the data for focusing on uninterpreted areas. Other seismic attributes from the same seismic study, such as fault probability, can be opaquely displayed on the horizons. This would express the fault probability along a horizon surface.

Advanced light effects can be achieved by extending the incremental filtering used for creating shadows during volume rendering. For instance, instead of casting shadows, we can let the horizons cast their colors, thereby simulating emissively glowing horizons. Emissiveness can be used to expressively highlight important horizons embedded in the seismic volume or to distinguish them from other horizons (Figure 7g-i). Compared to other focus+context techniques, our rendering method gives focus by using inherent and natural properties of the lighting model.

5.3 Interaction

We designed the system with the focus on simple interaction that supports easy and rapid interpretation of horizons. Intuitive interaction with the seismic volume is achieved by using the leaf surface-id component. This gives the ability to select leaf-surfaces by clicking directly on the volume rendered image. When a leaf-surface is selected, its surface hierarchy is available and can be navigated. This allows the user to iteratively select higher-level surfaces until the optimal size is found.

Volumetric horizon selection is achieved by first storing the 2D screen-space coordinate of the point the user selects. While performing front-to-back compositing of volume slices during volume rendering, the first hit with a nontransparent voxel is detected at the 2D coordinate on the compositing slice. The corresponding leaf surface-id is fetched from the leaf surface-id volume. This id identifies the selected leaf-surface and its color mapping is changed from full transparency to a color indicating it is selected.

Due to the dense nature of seismic data, we have defined a roaming box which confines the volume rendering. The user can easily move the roaming box through the volume and select horizons by clicking on the sides of the roaming box. Section 6 describes such an interaction scenario.

6 RESULTS

In this section we present a scenario of navigating and interacting for rapid horizon interpretation. In the scenario, two horizons are interpreted by using surface picking and hierarchical compositing of surface-parts into larger surfaces.

We use a white single-colored transfer function with low to high values mapped from transparent to opaque. Dark areas are due to shadows cast by opaque horizons through (semi)transparent space onto other horizons. Typically transfer functions for seismic-volume rendering use a black to white or red-white-blue gradient to easier discern horizons. With shadowing the color gradient specification is not necessary. This simplifies the transfer function setup thus supporting rapid interpretation.

To identify a horizon for interpretation, the side of the roaming box is moved through the volume. An interesting horizon is

identified on the front-side of the roaming box (Figure 6a) and the horizon is selected using point-and-click (red arrow in Figure 6a). This results in a selected leaf-surface shown in pink. Instead of continuing to select small leaf-surfaces along the horizon, the hierarchy information is utilized. By using the keyboard up-arrow all surfaces in the hierarchy level above the current one are colored in pink. The user can navigate up to the highest hierarchy level that yields a correct horizon and use it as a starting point of the horizon assembly. Figures 6b-f show increasing surface-hierarchy levels. Going from the hierarchy level in Figure 6e one level up as shown in Figure 6f, erroneous horizons above the current one are included. The error is revealed by the irregular and overlapping structures that do not exist in horizons. Overlapping wrong horizons are easy to spot by our method due to the shadows they cast on horizons directly below them. Therefore the user goes back to the previous level shown in Figure 6e and uses this as a starting point. He has now selected the horizon at the highest level in the hierarchy without errors. This surface constitutes a horizon with holes. The holes might consist of unselected surface-parts. The user attempts to fill these holes by selecting potentially segmented surface-parts located in the holes. To select these surfaces, the roaming-box side is moved to intersect the holes (Figure 7a). The red arrow shows where the user selects the horizons. In Figure 7b the resulting selected surface-part is shown in green. The process of moving the roaming-box side over the horizon and filling out holes where possible is performed over the entire horizon. The result can be seen in Figure 7c with five green surface-parts representing filled-in holes indicated with yellow arrows. Now one horizon has been interpreted. The user selects another horizon color (yellow) and repeats the procedure (Figure 7d). Two interpreted horizons can be seen in Figure 7e. Different rendering types are shown in Figures 7f-k. In Figure 7f the top horizon is shown semitransparently so the underlying data can be seen. In Figure 7g the top horizon with an emissive factor is shown. Emissiveness can be used for bringing attention to a certain horizon. Figure 7h shows the bottom horizon emissively. In Figure 7i both horizons are shown emissively. On the side face of the roaming box one can see how the emissiveness interacts with the volume rendering by casting light from the emissive object onto its surroundings. In Figures 7j and k the horizons extruding from an opaque cube are shown using a black to white transfer function. Ambient occlusion-like shading of volumes and horizons gives these renderings a quality comparable to a manually shaded illustration.

In Figure 7k a scenario with seismic data around a hypothetical boring well and its relation to the two horizons is shown. The interpretation just presented could easily be performed in less than ten minutes. This short interpretation time underlines the potential of our approach.

Implementation Details

The software has been developed using the VolumeShop [5] framework. For preprocessing the dataset of size 256^3 used in this article, about 2000 surfaces were grown and subdivided, constituting altogether about 2 million triangles. The hierarchy for each surface is stored in memory in a downstripped version containing only leaf surface-ids in the leaf nodes without the list of triangle-indices defining the surface-part. The size is then less than 1MB.

With unoptimized preprocessing, it took 1 hour to calculate the distance transform using a maximal distance of 10 voxels. The unoptimized brute-force surface-growing from each voxel in the volume took five hours and it took one hour for the hierarchy creation of the grown surfaces. We ran the preprocessing on a 1.8 Ghz AMD Athlon 64, Dual core processor 3800. Preprocessing could be sped up by using existing optimized distance transform algorithms running on GPUs. Growing could be parallelized by reimplementation in CUDA. However, the preprocessing is very labor intensive and seismic volumes are increasing in size. We do not expect to be able

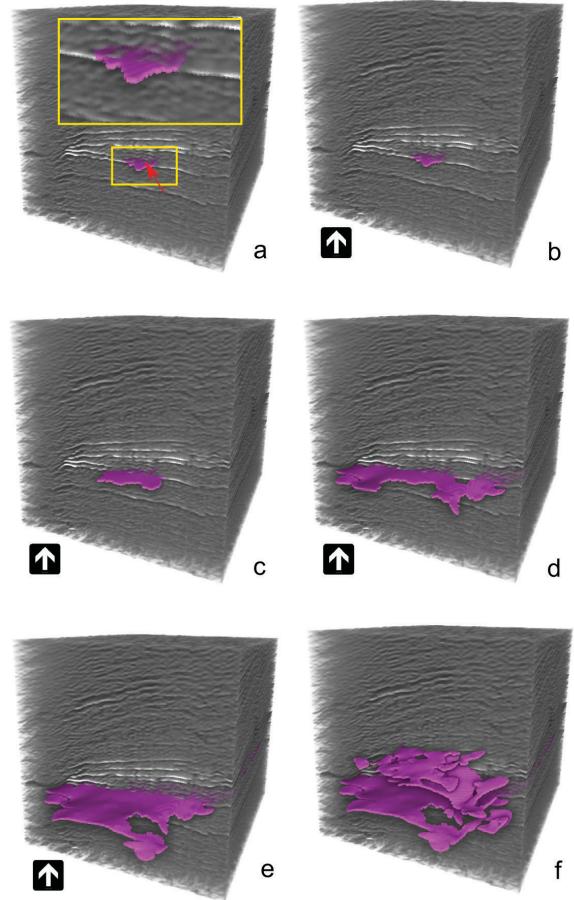


Figure 6: Selecting a leaf-surface (a) and navigating stepwise up in the hierarchy (b-f) until an erroneous surface is shown (f).

to perform preprocessing on the order of minutes in the near future. A realistic scenario would be to settle for processing times taking less than one night, thus having the data ready the next day.

For rendering we used an Nvidia GeForce 275. The 3-component volume was represented with one byte for the seismic value, one byte for the distance and two bytes for the leaf surface-id taking 64MB, thus 4 times larger than the original seismic volume. During interaction we achieved 25 FPS in a 500 by 500 window. The high-resolution images of size 1000×1000 used in this article were taken with 8 samples per voxel resulting in a speed of 7-14 FPS.

7 CONCLUSION

We have presented a system for rapid interpretation and expressive visualization of seismic horizons by carefully combining appropriate technologies. Our main contributions are horizon growing with surface subdivision, implicit horizon representation enabling single-pass advanced rendering and using an illumination model supporting emission for clearer visualization of noisy acoustic seismic reflectance data.

8 ACKNOWLEDGEMENTS

Data used in this article is by courtesy of Norske ConocoPhillips AS, ExxonMobil Norway AS, Total E&P Norge AS, Statoil AS and Petoro AS. The work was performed partly under the GeoIllustrator project sponsored by Statoil and Christian Michelsen Research.

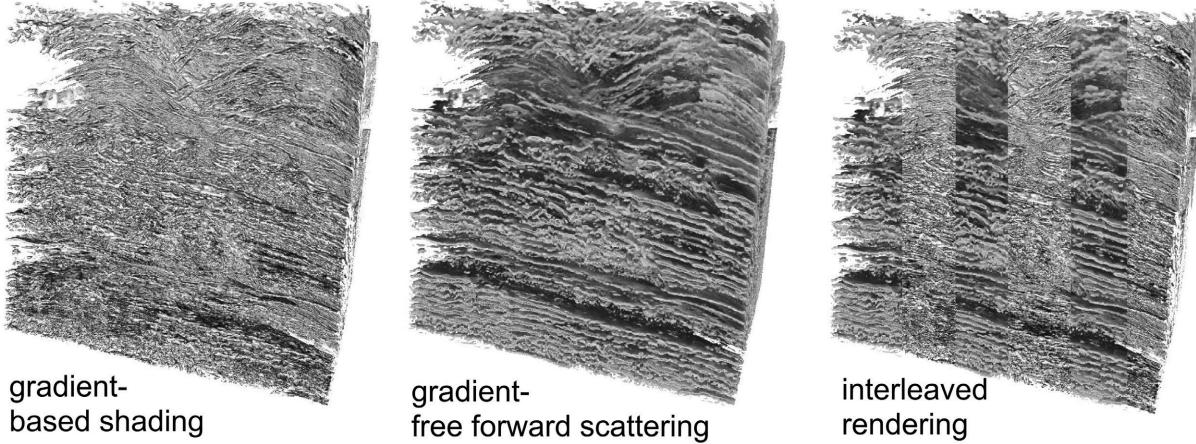


Figure 5: A comparison of gradient (left) and gradient-free shading (middle). In the right image, the two left images are interleaved and one can see how 3D and depth perception is improved and high frequency noise reduced in the gradient-free method.

REFERENCES

- [1] M. Attene, M. Spagnuolo, and B. Falcidieno. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, March 2006.
- [2] Autodesk AutoCAD, <http://www.autodesk.com/>, 2009.
- [3] U. Behrens and R. Ratering. Adding shadows to a texture-based volume renderer. In *Proc. of IEEE Symposium on Volume Visualization 1998*, pages 39–46, 1998.
- [4] H. Borgos, O. Gramstad, G. Dahl, P. L. Guern, and L. Sonnenland. Extracting horizon patches and geo-bodies from 3d seismic waveform sequences. *SEG/New Orleans Annual Meeting*, pages 1595–99, 2006.
- [5] S. Bruckner and M. E. Gröller. Volumeshop: An interactive system for direct volume illustration. In H. R. C. T. Silva, E. Gröller, editor, *IEEE Visualization 2005*, pages 671–678, Oct. 2005.
- [6] S. Bruckner and M. E. Gröller. Enhancing depth-perception with flexible volumetric halos. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1344–1351, 2007.
- [7] 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.
- [8] P. Desgranges, K. Engel, and G. Paladini. Gradient-free shading: A new method for realistic interactive volume rendering. In *Proc. of Vision, Modeling, and Visualization 2005*, pages 209–216, 2005.
- [9] M. Faraklouli and M. Petrou. Horizon picking in 3D seismic data volumes. *Machine Vision and Applications*, 15(4):216–219, 2004.
- [10] Google SketchUp, <http://sketchup.google.com/>, 2009.
- [11] F. Hernell, P. Ljung, and A. Ynnerman. Local ambient occlusion in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 99(2), 2009.
- [12] F. Hernell, A. Ynnerman, and P. Ljung. Efficient ambient and emissive tissue illumination using local occlusion in multiresolution volume rendering. In *Proc. of Volume Graphics 2007*, pages 1–8, 2007.
- [13] A. Iske and T. Randen, editors. *Atlas of 3D Seismic Attributes, Mathematics in Industry, Mathematical Methods and Modelling in Hydrocarbon Exploration and Production*. Springer, 2006.
- [14] M. W. Jones. 3d distance from a point to a triangle. *Technical Report CSR-5-95, Department of Computer Science, University of Wales Swansea*, February 1995.
- [15] M. W. Jones, J. A. Baerentzen, and M. Sramek. 3d distance fields: a survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics (IEEE TVCG)*, 12(4):581–599, 2006.
- [16] J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson. A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):150–162, 2003.
- [17] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8:29–37, 1987.
- [18] 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.
- [19] K. Marfurt. Robust estimates of 3D reflector dip and azimuth. *Geophysics*, 71:P29, 2006.
- [20] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.
- [21] D. Patel, C. Giertsen, J. Thurmond, J. Gjelberg, and M. E. Gröller. The seismic analyzer: Interpreting and illustrating 2d seismic data. *IEEE Transactions on Visualization and Computer Graphics (IEEE TVCG)*, 14(6):1571–1578, Oct 2008.
- [22] D. Patel, C. Giertsen, J. Thurmond, and M. E. Gröller. Illustrative rendering of seismic data. In *Proc. of Vision Modeling and Visualization*, pages 13–22, 2007.
- [23] 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.
- [24] B. T. Phong. Illumination for computer generated pictures. *ACM Communications*, 18(6):311–317, 1975.
- [25] J. Plate, T. Holtkaemper, and B. Fröhlich. A flexible multi-volume shader framework for arbitrarily intersecting multi-resolution datasets. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1584–1591, 2007.
- [26] 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.
- [27] C. Rezk-Salama. Gpu-based monte-carlo volume raycasting. In *Proc. of Pacific Graphics 2007*, pages 411–414, 2007.
- [28] T. Ropinski, J. Meyer-Spradow, S. Diepenbrock, J. Mensmann, and K. H. Hinrichs. Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum*, 27(2):567–576, 2008.
- [29] T. Ropinski, F. Steinicke, and K. H. Hinrichs. Visual exploration of seismic volume datasets. *Journal Proc. of WSCG '06*, 14:73–80, 2006.
- [30] Schlumberger and Petrel. Petrel seismic interpretation software, schlumberger information solutions (sis).
- [31] M. Schott, V. Pegoraro, C. Hansen, K. Boulanger, J. Stratton, and K. Bouatouch. A directional occlusion shading model for interactive direct volume rendering. *Computer Graphics Forum*, 28(3):855–862, June 2009.
- [32] A. Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.
- [33] P. Silva, M. Machado, and M. Gattass. 3d seismic volume rendering. *8th Intern. Congress of The Brazilian Geophysical Soc.*, 2003.
- [34] A. J. Stewart. Vicinity shading for enhanced perception of volumetric data. In *IEEE Visualization 2003*, pages 355–362, 2003.
- [35] R. Yagel, A. Kaufman, and Q. Zhang. Realistic volume imaging. In *IEEE Visualization 1991*, pages 226–231, 1991.

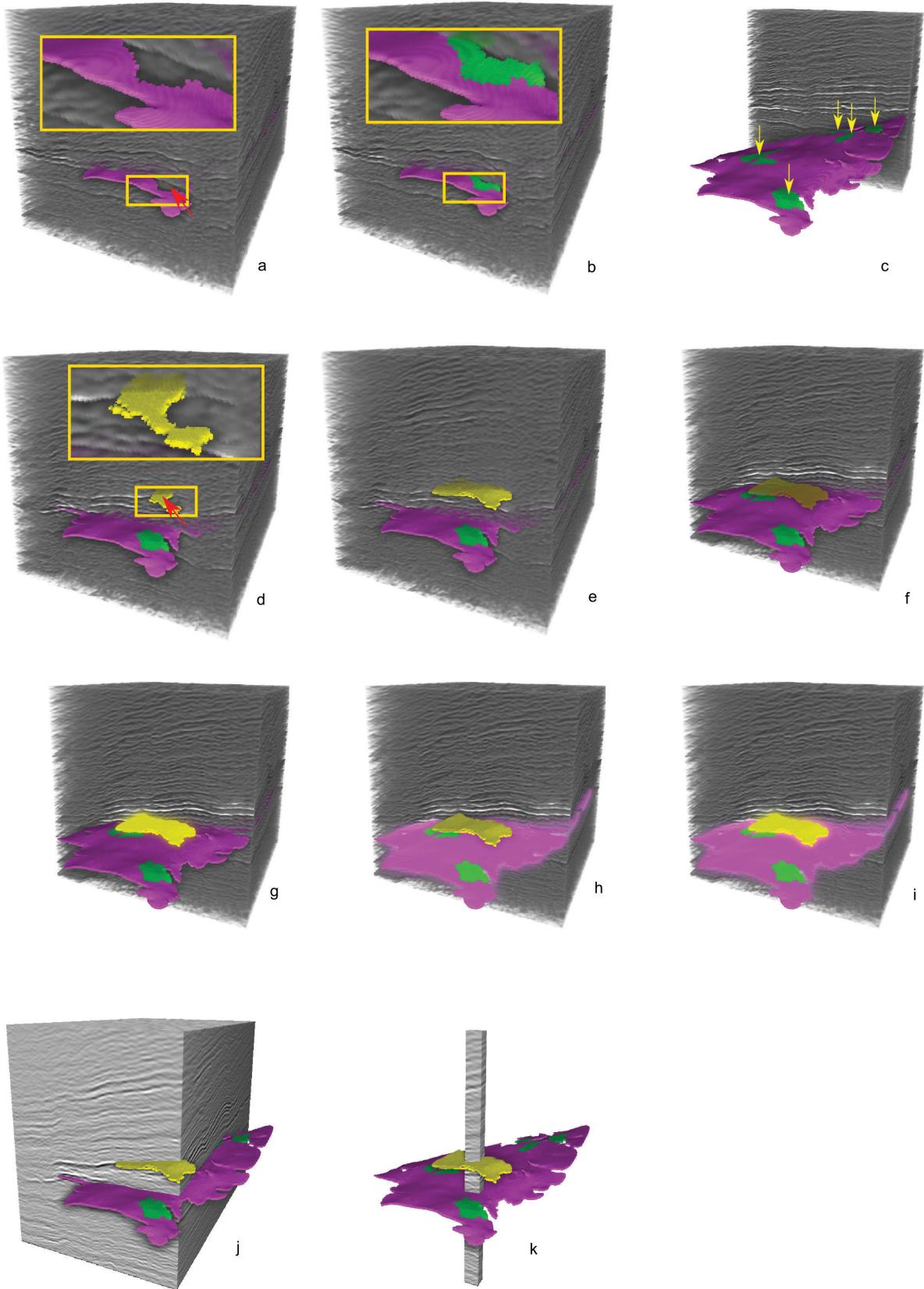


Figure 7: The steps of interpreting two horizons, filling out holes (a-c). Creating a new horizon (d-e). Top horizon semitransparent in (f) and emissive in (g). Bottom horizon emissive in (h). Both horizons emissive in (i). Opaque volume with different roaming boxes in (j) and (k).