



HAL
open science

VolumeExplorer: Roaming Large Volumes to Couple Visualization and Data Processing for Oil and Gas Exploration

Laurent Castanie, Bruno Lévy, Fabien Bosquet

► **To cite this version:**

Laurent Castanie, Bruno Lévy, Fabien Bosquet. VolumeExplorer: Roaming Large Volumes to Couple Visualization and Data Processing for Oil and Gas Exploration. IEEE Visualization 2005, Oct 2005, Minneapolis/USA, pp.247 - 254, 10.1109/VISUAL.2005.1532802 . inria-00105607

HAL Id: inria-00105607

<https://inria.hal.science/inria-00105607v1>

Submitted on 11 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

VolumeExplorer: Roaming Large Volumes to Couple Visualization and Data Processing for Oil and Gas Exploration

Laurent Castanié*
Earth Decision Sciences
Gocad Research Group

Bruno Lévy†
INRIA Lorraine
Project ALICE

Fabien Bosquet‡
Earth Decision Sciences

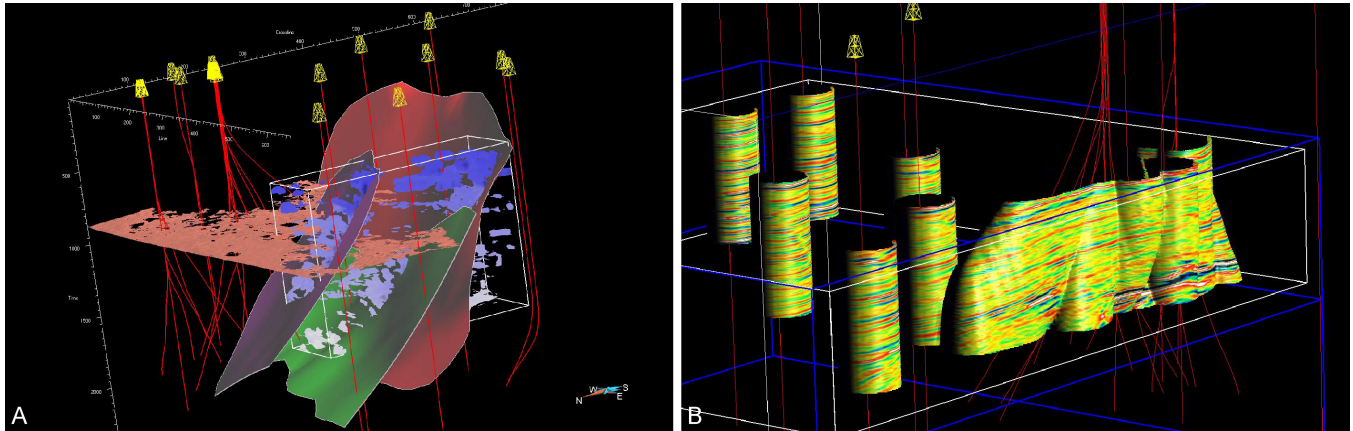


Figure 1: Volume visualization in 3D probes. (A) Volume rendering of high discontinuities related to the presence of faults. (B) Isosurface of a distance map to production wells painted with seismic amplitudes.

Abstract

In this paper, we present a volume roaming system dedicated to oil and gas exploration. Our system combines probe-based volume rendering with data processing and computing. The daily oil production and the estimation of the world *proven-reserves* directly affect the barrel price and have a strong impact on the economy. Among others, production and correct estimation are linked to the accuracy of the subsurface model used for predicting oil reservoirs shape and size. Geoscientists build this model from the interpretation of seismic data, i.e. 3D images of the subsurface obtained from geophysical surveys. Our system couples visualization and data processing for the interpretation of seismic data. It is based on volume roaming along with efficient volume paging to manipulate the multi-gigabyte data sets commonly acquired during seismic surveys. Our volume rendering lenses implement high quality pre-integrated volume rendering with accurate lighting. They use a generic multi-modal volume rendering system that blends several volumes in the spirit of the “stencil” paradigm used in 2D painting programs. In addition, our system can interactively display non-polygonal isosurfaces painted with an attribute. Beside the visualization algorithms, automatic extraction of local features of the subsurface model also take full advantage of the volume paging.

*e-mail: castanie@earthdecision.com

†e-mail: bruno.levy@loria.fr

‡e-mail: bosquet@earthdecision.com

CR Categories: I.3.8 [Computer Graphics]: Applications J.2 [Computer Applications]: Physical Sciences and Engineering—Earth and Atmospheric Sciences D.4.2 [Operating Systems]: Storage Management I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing Algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

Keywords: Oil and gas exploration, seismic interpretation, large volumes, volume bricking, out-of-core, volume roaming, paging, texture-based volume visualization, multi-modal rendering, programmable graphics hardware

1 Introduction

Seismic interpretation is an important task in the oil and gas exploration-production (EP) workflow [9, 26, 21]. From a 3D voxel image called *seismic volume*, geophysicists and geologists build a surfacic model of the subsurface. It is composed of horizons that represent the horizontal interfaces between the major geological layers, and faults that represent vertical discontinuities resulting from the geological history of the sedimentary basin. Horizons and faults basically outline possible traps for the precious hydrocarbons. A precise understanding of this model is therefore necessary for production wells planning and reserves estimation.

Visualizing seismic volumes is particularly challenging due to the constant growth of data sets size, commonly reaching several gigabytes of data. Seismic data are very noisy and have high spatial frequencies, making it very difficult to provide meaningful volume images. Classical full volume rendering methods used in oil and gas EP result in cluttered inexploitable images. Most papers dealing with the visualization for oil and gas EP have focused on the problem

of huge data sets visualization [9, 26, 21]. These methods use a single roaming approach based on local volume rendering lenses along with efficient volume paging. However, we believe that the main challenge in seismic interpretation is to implement efficient **qualitative** tools to help the extraction of **quantitative** information. We therefore couple data analysis tools with data processing tools to enable visualization-driven feature extraction. This provides an integrated application where visualization as well as processing algorithms take full advantage of volume paging.

1.1 Related Work

The implementation of an interpretation package for large seismic data faces two main problems in visualization: visualizing large volumes and implementing high quality volume rendering algorithms to overcome the inherent noisy and cluttered nature of seismic data.

Large Volumes: Dealing with multi-gigabyte volumes is difficult, due to the required size of both texture memory and main memory, and due to the limited system data bandwidth. Volume bricking is a common approach that makes it possible to stream smaller texture bricks through the rendering system and render them in a back-to-front order [28]. An octree-based multi-resolution approach [19, 33] saves AGP bandwidth (i.e. bandwidth between main memory and texture memory) where full resolution is not mandatory. Nguyen and Saupe [25] use wavelet-based compression to make levels of detail fit into main memory. Such compression along with multi-resolution [11] achieves interactive full volume rendering of multi-gigabyte data sets.

However, high resolution variations of seismic data induce cluttered images with classical full volume rendering. Roaming small volume rendering lenses is more suitable for seismic interpretation [21] and allows efficient paging strategies. The OpenGL Volumizer library [1] combines multi-resolution with volume roaming. Plate *et al.*'s Octreemizer system [26] for seismic visualization achieves interactive frame rates with data sets of several gigabytes in size. Both systems trade texture resolution for rendering speed when the user moves inside the volume in order to keep reactive.

Volume Visualization: We will focus on direct volume visualization methods such as volume rendering. Volume rendering considers the volume as a semi-transparent space and applies high quality lighting models [23]. Its first application to seismic data was proposed by Sabella [32].

Volumes are rendered using back-to-front semi-transparent sampling slices [18]. This approach benefits from 2D texture mapping hardware acceleration [2, 27]. The introduction of 3D texture [4] made interactive view-aligned slices possible thanks to hardware accelerated trilinear interpolation. This has been improved to achieve non-polygonal shaded isosurfaces rendering [35] and to enable diffuse illumination in volume rendering [8, 24].

In classical volume rendering, high quality requires high sampling rates [27], which results in heavy performance losses. Williams *et al.* have proposed high accuracy optical models for cell projection algorithms [36]. They were improved by Roettger *et al.* [31] and generalized to texture-based volume rendering by Engel *et al.* [10]. Instead of rendering textured slices, they basically render slabs with pre-integrated ray integrals. This achieves high quality volume rendering without requiring to increase the sampling rate. Roettger and Ertl accelerate accurate pre-integration with 2D texture hardware [29], while Lum *et al.* [20] propose a faster algorithm. The latter also introduce an interpolated pre-integrated lighting to remove artifacts at slab

transitions. However, to our knowledge, these techniques have never been used in oil and gas EP, mainly relying up to now on the non-programmable GPUs used by the heavy industry. We show in this paper how this domain can benefit from such methods.

Multimodal data: Seismic interpretation often benefits from the combination of multiple volumes [22, 7]. In the classification of multivariate data by Kaufman and Mueller [13], such data is called *multimodal* as opposed to *multichannel* in the case of RGB color volumes. Kniss *et al.* propose the use of multidimensional transfer functions [14] for multivariate volume analysis in general.

For multimodal volumes, Cai and Sakas [5] propose an intermixing approach and define three levels of intermixing, depending on when the data are mixed in the volume rendering pipeline. Volumetric clipping introduced by Weiskopf *et al.* [34] is a particular case of accumulation-level intermixing defined by Cai and Sakas. Volumetric clipping has been adapted to pre-integrated volume rendering by Roettger *et al.* [30].

Finally, Jen *et al.* [12] use painted isosurfaces to combine two scalar volumes in the exploration of confocal microscopy data obtained from the brain. However, they use indirect methods to actually extract surface representations out of the volume data instead of interactively visualizing non-polygonal isosurfaces. We will show how accumulation-level intermixing can interactively paint non-polygonal isosurfaces (Section 4.3). We use a multimodal rendering system inspired by the “stencil” paradigm used in 2D painting.

1.2 Contributions and Overview

In this paper, we present a probe-based rendering system that provides both 2D and 3D probes along with volume paging for efficiently roaming large volumes (Section 3). This system is based on OpenGL and has been integrated into GOCAD, a CAD software dedicated to geosciences and widely used by the oil and gas EP industry.

Our main contributions to the domain are:

- Integrating high quality GPU-based volume rendering algorithms in a seismic interpretation package. We show how seismic data analysis benefits from such visualization techniques (Section 4.1).
- Introducing a new generic multimodal volume rendering system. Our system blends several volumes in the spirit of the “stencil” paradigm used in 2D painting programs. This enables multimodal correlation between different attributes (Section 4.2). Coupled with non-polygonal isosurfaces rendering algorithms, it makes it possible to render painted isosurfaces interactively (Section 4.3).
- Coupling data processing tools with a probe-based volume rendering system. Using our tools, the data extraction process can be steered by visualization. Moreover, a single cache is used for both visualization and data processing algorithms (Section 5).

A typical interpretation workflow in the context of oil and gas EP will be presented (Section 2), as well as a subsurface model resulting from the interpretation of a large data set following this workflow and using our system (Section 6).

2 Seismic Interpretation in Oil and Gas Exploration-Production

Seismic interpretation is at the beginning of the oil and gas EP workflow. Seismic data are stored in regular grids representing 3D images of the subsurface with a resolution of several tens of meters in each dimension. Acoustic waves are produced on the Earth's surface, they propagate through the geological layers, reflect on the geological interfaces and their amplitude is recorded by surface receivers. Their arrival time gives the depth of the reflecting interfaces, which makes it possible to build a 3D image of the subsurface. This image is stored in a 3D regular grid that contains the seismic amplitude values. Out of this 3D regular grid, geophysicists and geologists build a surfacic model called the *structural model*. It contains a series of horizons that represent the horizontal interfaces between the major geological layers and faults that correspond to vertical discontinuities resulting from geomechanical constraints in the sedimentary basin. They are respectively stored as 2D height fields and triangular surfaces. Possible hydrocarbon traps are predicted after a precise understanding of the structural model, which makes it possible to design the path of production wells with a high precision.

However, geophysical surveys are covering larger and larger areas. This has mainly two consequences: (1) From the software engineer point of view, this means that the seismic data sets generated by these surveys are growing continuously, commonly yielding several gigabytes of data. (2) From the geoscientist point of view, this means that the amount of information is becoming bigger, making it more difficult to filter the meaningful part. Seismic data often cover an entire sedimentary basin that contains several reservoirs.

Faced with this complexity, interpreters adopt various strategies, as exposed in [7]:

- **Micro-Interpretation:** This approach consists in directly interpreting small scale objects (hundreds of meters). A first step consists in finding economically interesting reservoirs and analysing them in terms of shape and size. Then, reservoirs are correlated at the basin scale to have a better idea of their regional organization.
- **Macro-Interpretation:** In this case, a first step investigates the entire basin for getting a global knowledge of its structure (several kilometers). Information is filtered and isolated. The user then analyzes local reservoirs to define their economical potential.

There is no preferred approach, both of them having its advantages and disadvantages. Most of the time, a combined strategy with a feedback between micro and macro filters is preferable to converge to a solution. The typical interpretation workflow is presented in Figure 2. Starting from seismic volume data, this workflow outputs a structural model along with a set of production wells (Figure 9). Note that production wells depend on additional steps such as property modeling and flow simulations in the subsurface model, that are beyond the scope of this paper.

Each feature of the system presented in the remainder of the paper contributes to specific steps of the workflow:

- **Global overview:** A first global approach of the volume is necessary to identify potential reservoirs. It is based on macro-interpretation tools such as the *Slicer* and *Section* probes (Section 3).

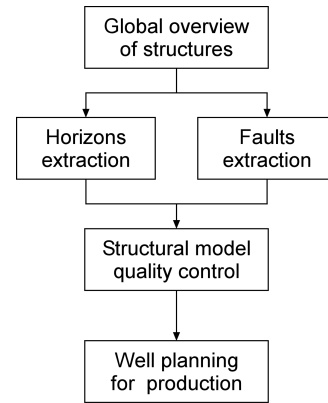


Figure 2: Typical interpretation workflow in oil and gas EP.

- **Horizons extraction:** Then, interpreters “dive” into the specific features observed at the global scale, typically roaming *Skin* or *Volume* probes (Section 3) to observe relevant horizons and pick initial seed points. From these seeds, horizon auto-picking algorithms extract 2D height fields that represent the geological horizons (Section 5).
- **Faults extraction:** Faults signatures are captured with high quality volume rendering of seismic amplitudes (Section 4.1). Beside, fault extraction highly benefits from the computation of additional information such as the semblance cube (Section 5). Semblance volume rendering identifies lateral discontinuities related to faults. After filtering, cloud points related to faults can also be extracted from semblance values.
- **Quality control:** A coherent structural model is based on a macro-topology that defines the relations between horizons and faults. Isosurfaces of distance maps to horizons and/or faults painted with seismic amplitudes (Section 4.3) are helpful in understanding the relation between these geological surfaces. Note that distance maps can be computed in local regions such as 3D probes to avoid multi-gigabyte computations when they are not mandatory.
- **Well planning:** Finally, the planning of production wells assumes a precise knowledge of the structural model as well as of the nature of the geological layers to be traversed. Isosurfaces of distance maps to production wells painted with seismic amplitudes (Section 4.3) dramatically make this step easier.

3 Volume Roaming for Oil and Gas Exploration-Production

We implemented both 2D and 3D probes along with volume paging for micro and macro interpretation. 2D probes are typically dedicated to macro-interpretation while 3D probes are more suitable for micro-interpretation.

2D Probes for Macro-Interpretation: As previously mentioned in the introduction, full volume rendering at a macro scale is useless for seismic data visualization. 2D probes are therefore preferable for macro-interpretation. We defined three types of 2D probes:

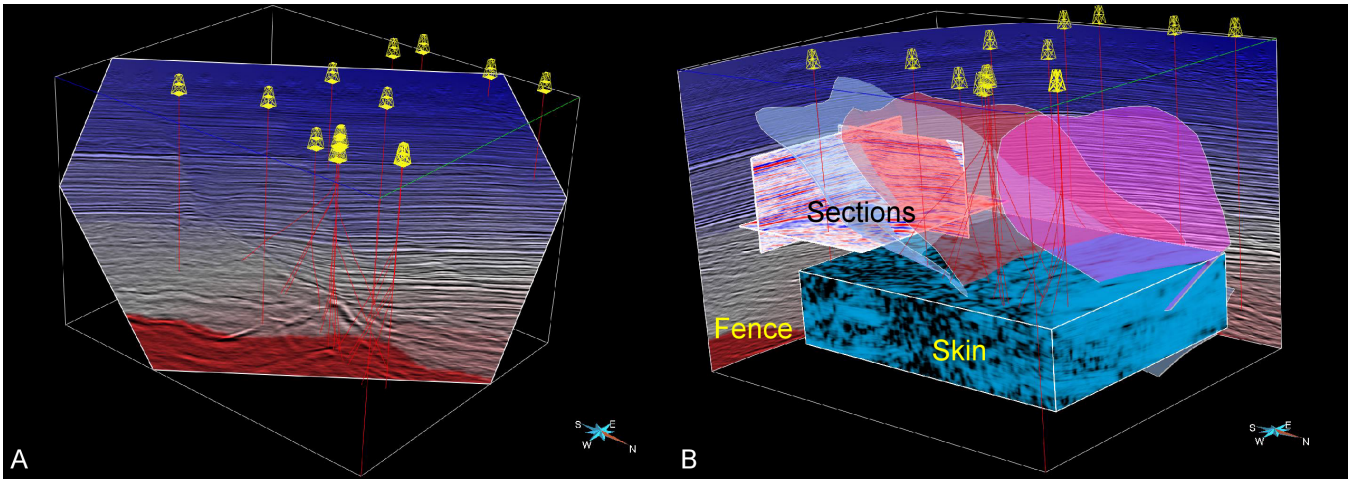


Figure 3: 2D texture based visualization with probes. (A) *Slicer* probe with production wells. (B) *Section*, *Fence* and *Skin* probes with production wells and semi-transparent fault surfaces. *Slicer* and *Fence* probes are painted with seismic amplitudes (grey scale) and interval velocity (blue-white-red), *Section* probes with seismic amplitudes and *Skin* probe with semblance (low values in black, see Section 5).

- *Slicer* probe: This is the natural tool for macro-interpretation. The slicer is an arbitrary slice in the entire volume centered on the center of projection of the viewing camera (i.e. on the rotating center of the model) and facing the viewer (Figure 3-A). During user interaction, the slicer keeps fixed while the volume turns around.
- *Section* probes: They are defined as slices of arbitrary extension orthogonal to one volume axis (Figure 3-B).
- *Fence* probes: A series of vertical pillars with the same arbitrary extension define successive vertical 2D panels that form a fence (Figure 3-B). They are typically used for correlating several reservoirs in the whole basin or several wells in a model.

Such 2D probes take advantage of volume paging into main memory. However, to reduce GPU memory usage and keep it available for 3D probes, they are implemented with 2D textures. This is motivated by the fact that they are most of the time dedicated to macro-interpretation, while volume paging is most efficient with volume roaming, i.e. with small 3D probes.

3D Probes for Micro-Interpretation: They define small parallelepipedic lenses of arbitrary size roamed around the volume and dedicated to micro-interpretation. We defined three rendering types for 3D probes:

- *Skin* probes: The simplest way of rendering 3D probes is to draw their boundary painted with attributes (Figure 3-B). This rendering mode does not use much resources as we use three 2D textures to draw the visible faces.
- *Volume* probes: They implement 3D texture based volume rendering with back-to-front view-aligned slices and pre-integrated transfer function [10, 20] (Figures 1-A and 4-B, and Section 4.1).
- *Isosurfaces* probes: They render multiple non-polygonal shaded isosurfaces [35, 10] with the same

proxy geometry as in *Volume* probes. In our application, such rendering mode is particularly powerful when embedded in a multimodal system to display painted isosurfaces (Figure 1-B and Section 4.3).

In the remainder of the paper, we will focus on *Volume* and *Isosurfaces* probes, as they actually implement volume visualization tools and take full advantage of volume paging.

Volume Paging: To optimize roaming with probes, we implemented a volume paging system. Note that in the oil and gas industry, visualization needs to be complemented with data analysis (Sections 4 and 5). Our roaming system is based on a brick decomposition of the volume to exploit the spatial locality of both visualization tasks (i.e. 3D probes roaming) and general-purpose algorithms (i.e. semi-automatic feature extraction algorithms). We have implemented a two-level cache between disk, main memory and texture memory. In this system, general-purpose algorithms and 2D texture based visualization only use the first level of cache (i.e. main memory cache) while 3D texture based visualization uses the texture memory cache as well. Both cache levels adopt a classical LRU strategy to determine which brick should be removed when the cache is full.

Although pure visualization only needs 8-bit data, data processing algorithms require full precision. We therefore keep high precision in the main memory cache and convert data to 8-bit when transferring to texture memory. We store textures in OpenGL `GL_RGBA` format: RGB color channels contain a pre-computed gradient for lighting computation and the opacity channel contains the sample value.

To improve interactivity, classical techniques use a multi-resolution representation of the volume. However, this cannot be done in our specific case. Due to the high vertical frequency of seismic data (Figure 3-A), multi-resolution rendering would result in blurry useless images. For this reason, as classically done in the oil and gas visualization domain, to maintain the system reactivity during user motion, volume rendering is done in the front buffer. User interaction can interrupt the current rendering, and display only the 2D probes and the outline of volumic probes while the viewpoint is changing. When the user stops moving the mouse, volume rendering is re-started in the volumic probes.

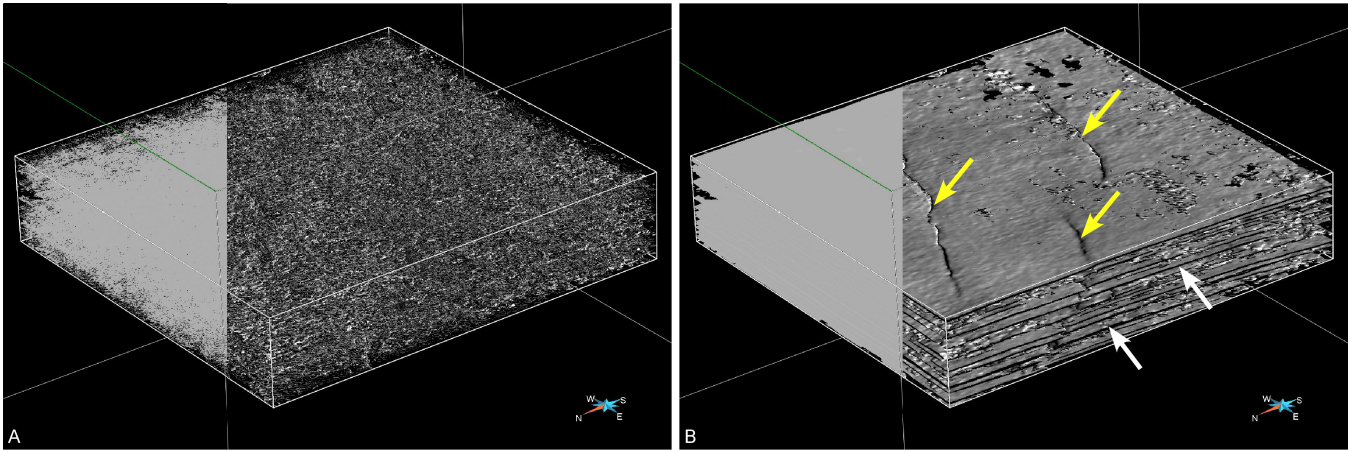


Figure 4: Volume rendering of seismic amplitudes in a 3D probe of size 480x480x510 (roaming within a volume of size 2608x661x811, 32 bits per voxel) with a transfer function containing a series of spikes in the moderate amplitudes. Left sides and right sides are respectively rendered without and with lighting. (A) Classical slicing-based (post-classification) volume rendering with 1000 slices. (B) Pre-integrated volume rendering with 600 slices. (Yellow and white arrows respectively show faults and horizons). Note how the “layer-cake” structure is revealed in (B), whereas it is completely invisible in (A).

4 Data Visualization and Data Analysis

Classical volume rendering on seismic data usually results in useless cluttered images. Moreover, seismic data is often multimodal. It contains additional volumes of information derived from the initial one via signal processing algorithms (Section 5) or from already extracted information such as distance maps to a set of horizons and faults. This section shows how seismic interpretation can benefit from high quality volume rendering and presents a generic multimodal system that enhances the data analysis workflow.

4.1 High Quality Volume Rendering

In classical slicing-based volume rendering techniques [18, 2, 27], high rendering quality is achieved at the price of high sampling rates (i.e. low performance). High quality volume rendering uses *slabs* instead of *slices*. It is based on a pre-integration of the volume rendering integral for all possible couples of front and back slice sample values. It was first introduced for volume rendering tetrahedral meshes [36, 31] and further generalized to texture-based volume rendering by Engel *et al.* [10]. For fast accurate pre-integration of the transfer function, we use the incremental subrange algorithm introduced by Lum *et al.* [20]. Besides, pre-integrated volume rendering with lighting is problematic as lighting normals are not taken into account during pre-integration. Arbitrarily sampling the normal in the ray segment at rendering time to compute lighting produces artifacts on slab transitions. We rather use the interpolated pre-integrated lighting introduced by Lum *et al.* [20] that uses four pre-integrated lookup tables instead of the initial one.

To implement high quality volume rendering for seismic data, we need to consider the specificity of the data. As shown in Figure 5, data values in a seismic volume usually have a gaussian distribution. According to Brown [3], the relevant information on the structural organization of geological layers is contained in the moderate amplitudes, i.e. between the mean and the extrema of the distribution. Seismic interpreters therefore focus on narrow ranges of seismic amplitudes, which requires specific transfer functions most of the time containing a series of spikes. In this

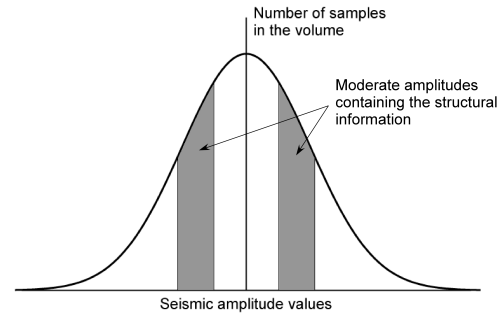


Figure 5: Typical gaussian distribution of seismic values. Moderate amplitudes correspond to relevant structural information.

case, classical slicing-based volume rendering fails to capture the relevant structures in the volume as it misses the high frequencies in the transfer function. Note that neither supersampling nor lighting computations improve the result (Figure 4-A). In contrast, pre-integrated volume rendering, as shown in Figure 4-B, does not suffer from such singular transfer functions. Combined with lighting, it perfectly reproduces the spatial organization of geological structures. Pseudo-vertical faults, which trace is visible on the top face of the probe (yellow arrows), clearly affect the geological horizons (white arrows). From a historical point of view, the visualization systems dedicated to seismic interpretation are super-computers most of the time limited to OpenGL 1.1 or 1.2. They provide alpha blending and achieve good rendering quality at the price of being limited to smooth transfer functions and requiring high sampling rates. With classical volume rendering, transfer functions that contain a series of spikes require an infinity of sampling planes to achieve the same quality as pre-integrated volume rendering. However, today off-the-shelf PCs equipped with powerful GPUs that expose new functionalities are progressively adopted by oil and gas companies. This new trend gives us the opportunity of supplying seismic interpreters with high quality rendering techniques such as pre-integrated volume rendering.

4.2 Generic Multimodal System

Seismic interpretation often generates multiple volumes of data [22, 7]. For instance, interpreters compute distance maps to a series of extracted surfaces or to a series of wells. Beside structural information, additional volumes can be derived from the initial seismic data such as the semblance attribute (Section 5). To facilitate the interpretation of these multiple volumes, we provide a combined visualization with a generic multimodal system based on the same idea as the “stencil” paradigm in 2D painting programs.

Multimodal Interface: The system is based on the notion of *layer*. In 2D painting systems, 2D layers are stacked along an abstract third dimension. In a layer-based 3D volume visualization system, the volume layers are stacked along an abstract fourth dimension. Each volume to combine defines a layer with an associated 1D transfer function. We will show the versatility of our approach. For instance, as shown further, our system can be used to display non-polygonal isosurfaces painted with an attribute (Section 4.3). Three types of layers are defined and determine three possible ways for the user to interact with the transfer function. In our experiments, we have noticed that providing the user with the following three types of layers eases the multimodal visualization process (Figure 6):

- *Color*: RGBA color channels are freely edited in the transfer function.
- *Opacity*: RGB color channels are fixed to one (white color), while A can be modified.
- *Intensity*: A given color is defined for the layer and its intensity is modulated with either a white or a black background along the transfer function. Opacity A is fixed to one.

Multimodal Implementation: In contrast with the “stencil” paradigm, we do not combine final images, but rather combine layers at the fragment level. As defined by Cai and Sakas [5], it corresponds to accumulation-level intermixing. We use the *fragment program* OpenGL extension to implement both pre-integrated volume rendering used by each layer and the compositing of the layers. Note that each layer’s transfer function is pre-integrated independently, which means that ray segment pre-integration is done with image-level intermixing [5, 13]. Accumulation-level intermixing in ray integration would result in non-practical $2 * \text{number of channels}$ dimensional pre-integrated tables.

One typical application of this system is volumetric clipping [34]. The neighborhood of an horizon, a fault or a set of well paths often contains meaningful information. Assuming that a distance map to this particular object has been computed, it can be bound to an *Opacity* layer to play the role of a clipping volume.

4.3 Painted Isosurface Rendering

This system is highly versatile and can be easily exploited by the user to achieve non-standard visualization methods. For instance, it can be used to implement non-polygonal isosurfaces rendering algorithms. Thus, our system can interactively display isosurfaces painted with an attribute. Figure 1-B shows an isosurface of a distance map to a set of production wells painted with the seismic data. This provides both interactive extraction and painting of isosurfaces. To our knowledge, previous works that correlate multiple volumes with painted isosurfaces [12] extract the isosurface with explicit techniques such as the Marching Cubes. In contrast,

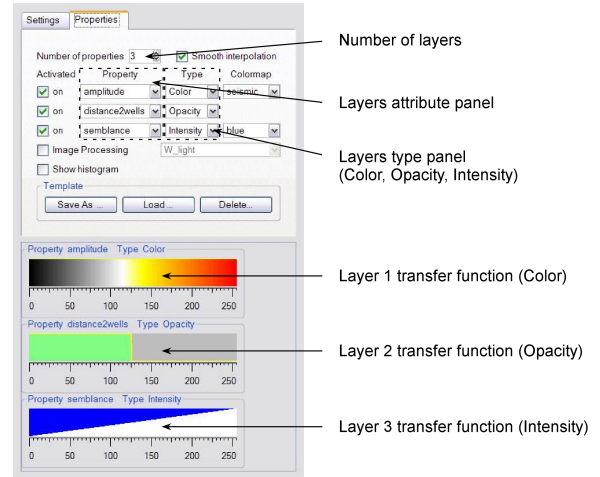


Figure 6: Layers editing in the multimodal system user interface.

our volume rendering based isosurface algorithm does not require any explicit extraction of the isosurface. As a result, better interactivity is obtained. We render multiple semi-transparent isosurfaces with a pre-integration approach as described in [10]. Note that we actually render “thick” isosurfaces to avoid holes [15].

5 Data Processing

Defining seismic data as 3D images is somewhat restrictive. In fact, each vertical array of values in the data volume corresponds to a set of seismic amplitudes sampling a sinusoidal signal f . We project the signal onto a basis of trigonometric polynomials in vertical windows defined over $[-w\delta t, +w\delta t]$ intervals. The projected signal f is given by:

$$f(t) = a_0 + \sum_{k=1}^m \left(a_k \cdot \cos\left(\frac{k\pi t}{w\delta t}\right) + b_k \cdot \sin\left(\frac{k\pi t}{w\delta t}\right) \right) \quad (1)$$

where δt is the width of a voxel along the t vertical axis, a_0, \dots, a_m and b_1, \dots, b_m are the coefficients of the projection and m is the degree of the polynomial ($m \leq w$, in practice $m = w$ and $w = 10$). Based on this projection, we provide some data processing algorithms to automatically extract horizons and faults. The complete description of the method is beyond the scope of this paper and is detailed in [17]. The algorithms re-use the cache system used by the visualization system.

Horizon Auto-Picking: A geological horizon is defined as the boundary between two geological layers. Acoustic waves are reflected on horizons, which produces strong continuous horizontal signatures on the seismic image. Given an original seed corresponding to a point on an horizon in the seismic volume, this signature can be automatically followed laterally. Our detection algorithm uses the projection of the seismic signal given in Equation (1) and the correlation function R_{fg} between two trigonometric polynomials f and g defined as follows:

$$\forall \tau \in [-w\delta t, +w\delta t], R_{fg}(\tau) = \frac{C_{fg}(\tau)}{C_{ff}(0) \cdot C_{gg}(0)} \quad (2)$$

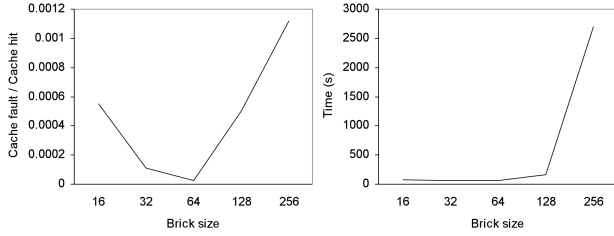


Figure 7: Cache efficiency relative to the brick size during horizon auto-picking in a 5.2 GB cube (2608x661x811, 32 bits per voxel). The cache size is 320 MB and the bricks are cubic. The extracted horizons are 661x811 height fields covering the entire cube. Left: Cache fault / cache hit. Right: Time of extraction in seconds.

where C_{fg} is the covariance between f and g defined by:

$$\forall \tau \in [-w\delta t, +w\delta t], C_{fg}(\tau) = \int_{-w\delta t}^{+w\delta t} f(t) \cdot g(t+\tau) dt \quad (3)$$

This semi-automatic extraction algorithm has a strong spatial locality that takes full advantage of the main memory cache described in Section 3. Figure 7 shows the performance of the cache relative to the brick size. We used the same workstation as in Section 6. Figure 8 shows the result of an horizon extraction along with the state of the main memory cache at the end of the extraction. For optimal visualization of horizons, we use view-dependent level-of-detail based on a quadtree data structure.

Fault Extraction: Faults are defined as pseudo-vertical geological singularities that laterally affect horizons. They are characterized on the seismic image by lateral discontinuities in the horizons (Figure 4-B). Their automatic extraction is based on the computation of an additional attribute called *semblance*. For each point of the volume, the semblance value is computed as a function of the signal at the point and at its n neighbors (e.g. $n = 8$). Assuming that f_1, \dots, f_{n+1} are the projected signals (see Equation (1)), semblance S is defined as follows:

$$S(f_1, \dots, f_{n+1}) = \frac{\int_{-w\delta t}^{+w\delta t} \left(\sum_{i=1}^{n+1} f_i(t) \right)^2 dt}{(n+1) \sum_{i=1}^{n+1} \int_{-w\delta t}^{+w\delta t} \left(f_i(t) \right)^2 dt} \quad (4)$$

Typically, semblance has high values where lateral continuity is preserved and low values where it is not. As a result, low values of semblance correspond either to faults (Figure 1-A) or to noise in the recorded signal. After filtering noise in the low values of semblance, we can extract point clouds corresponding to faults and triangulate them to generate fault surfaces. Beside semi-automatic extraction of faults, we also provide picking tools to extract them manually.

Computing a semblance cube generates a new volume stored with the same volume paging data structure as introduced in Section 3. Contrary to pure visualization packages, this means that our data structure does not only support read-only access but also read-write. In order to avoid updating disk data each time a brick is removed from memory, we maintain with each brick a “dirty” flag that specifies whether the concerned brick was modified.

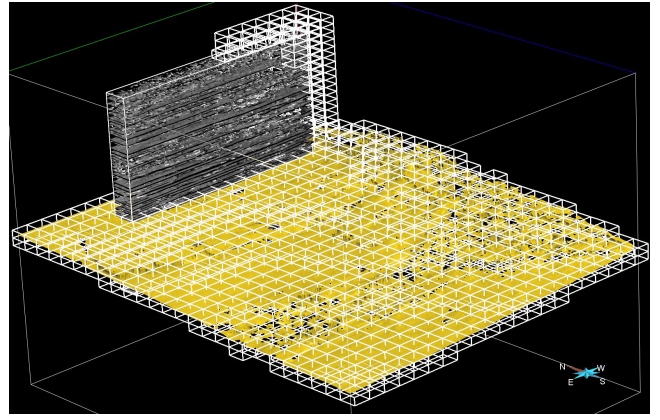


Figure 8: Horizon auto-picking resulting in the gold height field displayed with the memory cache (white bricks). The volume size is 5.2 GB (2608x661x811, 32 bits per voxel) and the cache size is 320 MB.

6 Results

Based on the workflow presented in Section 2 and using our system, we built the structural model and production wells presented in Figure 9. The corresponding seismic volume is a 32-bit 2608x661x811 cube of 5.2 GB (Figures 1, 3, 4 and 8). It contains four attributes: initial seismic data, semblance, another seismic property called the *interval velocity* and a distance map to the production wells. The distance map has been computed for the entire volume and stored as a new attribute. As a result, the entire data set has a size of 20.8 GB. The workstation for this study is a bi-Xeon 3.4 GHz with 3 GB of RAM. The graphics card is a Quadro FX 3400 with 256 MB of texture memory and a PCI Express bus. The main memory cache size was 1 GB (256 MB per attribute) and we used the entire 256 MB of the graphics card for the texture memory cache.

7 Conclusion and Future Works

We have presented a system dedicated to seismic interpretation for oil and gas exploration. This system is based on volume roaming along with volume paging to manage the large volumes of data commonly involved in oil and gas

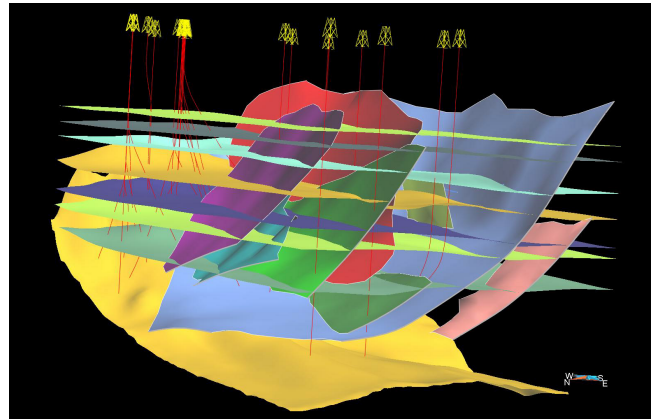


Figure 9: Structural model and production wells built out of a seismic cube of 5.2 GB (2608x661x811, 32 bits per voxel).

EP. It provides high quality volume rendering tools and is based on a versatile multimodal rendering system. Visualization is coupled with data processing algorithms to provide visualization-driven feature extraction. A single cache is used for both visualization and data processing. This system has been integrated into GOCAD, a CAD software dedicated to geosciences widely used by the oil and gas EP industry. Users feedback has been very positive until now.

In the near future, we plan to implement rendering optimizations such as empty space skipping and early-z rejection to accelerate volume rendering and non-polygonal isosurfaces [16]. We also plan to define other proxy geometries such as the horizon's quadtree to focus visualization on relevant features. From user feedback, we should implement n -dimensional transfer functions for either multimodal rendering or gradient magnitude versus scalar value rendering as in [14]. Finally, we have started the development of a distributed version of the algorithm for PC clusters [6].

8 Acknowledgements

Authors thank Earth Decision Sciences for supporting this work. Thanks to Jean-Laurent Mallet and the GOCAD consortium for providing hardware. The test workstation and graphics card were kindly provided by HP and NVIDIA.

References

- [1] BHANIRAMKA P. and DEMANGE Y. OpenGL Volumizer: A Toolkit for High Quality Volume Rendering of Large Data Sets. In *Proc. of IEEE Symp. on Volume Vis.*, pages 45–54, 2002.
- [2] BRADY M., JUNG K., NGUYEN H.T., and NGUYEN T. Two-Phase Perspective Ray Casting for Interactive Volume Navigation. In *Proc. of IEEE Visualization*, 1997.
- [3] BROWN A.R. *Interpretation of Three-Dimensional Seismic Data, Sixth Edition*. AAPG-SEG, Tulsa, 2004. 541 p.
- [4] CABRAL B., CAM N., and FORAN J. Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware. In *Proc. of IEEE Symp. on Volume Vis.*, pages 91–98, 1994.
- [5] CAI W. and SAKAS G. Data Intermixing and Multi-Volume Rendering. In *Proc. of Eurographics/IEEE Symp. on Visualization*, volume 19, pages 359–368, 1999.
- [6] CAVIN X., MION C., and FILBOIS A. COTS Cluster-Based Sort-Last Rendering: Performance Evaluation and Pipelined Implementation. In *Proc. of IEEE Visualization*, 2005.
- [7] CHRISTIE M. Thinking Inside the Box. *CSEG Recorder*, pages 50–57, June 2002.
- [8] DACHILLE F., KREEGER K., CHEN B., BITTER I., and KAUFMAN A. High-Quality Volume Rendering Using Texture Mapping Hardware. In *Proc. of Eurographics/SIGGRAPH Workshop on Graphics Hardware*, 1998.
- [9] DELANEY B. VizSim Technology Helps Find Oil Faster. *IEEE CG&A*, 19(2):10–16, March 1999.
- [10] ENGEL K., KRAUS M., and ERTL T. High-Quality Pre-Integrated Volume Rendering Using Hardware Accelerated Pixel Shading. In *Proc. of Eurographics/SIGGRAPH Workshop on Graphics Hardware*, pages 9–16, 2001.
- [11] GUTHE S., WAND M., GONSER J., and STRASSER W. Interactive Rendering of Large Volume Data Sets. In *Proc. of IEEE Visualization*, 2002.
- [12] JEN D., PARENTE P., ROBBINS J., WEIGLE C., TAYLOR R.M., BURETTE A., and WEINBERG R. Image Surfer: A Tool for Visualizing Correlations Between Two Volume Scalar Fields. In *Proc. of IEEE Visualization*, pages 529–536, 2004.
- [13] KAUFMAN A. and MUELLER K. *The Visualization Handbook*, chapter Overview of Volume Rendering. Elsevier, 2005.
- [14] KNISS J., KINDLMANN G., and HANSEN C. Multidimensional Transfer Functions for Interactive Volume Rendering. *IEEE TVCG*, 8(4):270–285, 2002.
- [15] KRAUS M. and ERTL T. *The Visualization Handbook*, chapter Pre-Integrated Volume Rendering. Elsevier, 2005.
- [16] KRÜGER J. and WESTERMANN R. Acceleration Techniques for GPU-Based Volume Rendering. In *Proc. of IEEE Visualization*, pages 287–292, 2003.
- [17] LABRUNYE E. *Extraction Automatique d'Information Géologique à Partir d'Images Sismiques Tridimensionnelles*. PhD thesis, INPL (Nancy), 2004.
- [18] LACROUTE P. and LEVOY M. Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transform. In *Proc. of ACM SIGGRAPH*, volume 28, 1994.
- [19] LAMAR E., HAMANN B., and JOY K.I. Multiresolution Techniques for Interactive Hardware Texturing-Based Volume Visualization. In *Proc. of IEEE Visualization*, 1999.
- [20] LUM E.B., WILSON B., and MA K.L. High-Quality Lighting and Efficient Pre-Integration for Volume Rendering. In *Proc. of Eurographics/IEEE Symp. on Visualization*, 2004.
- [21] MARSH A.J. Volume Roaming and Volume Visualization of Large 3D Datasets. In *Proc. of the Indonesian Petroleum Assoc. 28th Annual Meeting*, 2001.
- [22] MARSH A.J., KIDD G.D., and FURNISS A. 3D Seismic Visualization Using Multiple Volume Data Sets. In *Proc. of the Indonesian Petroleum Assoc. 27th Annual Meeting*, 2000.
- [23] MAX N. Optical Models for Direct Volume Rendering. *IEEE TVCG*, pages 99–108, 1995.
- [24] MEISSNER M., HOFFMANN U., and STRASSER W. Enabling Classification and Shading for 3D Texture Mapping Based Volume Rendering Using OpenGL and Extensions. In *Proc. of IEEE Visualization*, pages 207–214, 1999.
- [25] NGUYEN K.G. and SAUPE D. Rapid High Quality Compression of Volume Data for Visualization. In *Proc. of Eurographics/IEEE Symp. on Visualization*, volume 20, 2001.
- [26] PLATE J., TIRTASANA M., CARMONA R., and FRÖHLICH B. Octreemizer: A Hierarchical Approach for Interactive Roaming Through Very Large Volumes. In *Proc. of Eurographics/IEEE Symp. on Visualization*, 2002.
- [27] REZK-SALAMA C., ENGEL K., BAUER M., GREINER G., and ERTL T. Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization. In *Proc. of Eurographics/SIGGRAPH Workshop on Graphics Hardware*, pages 109–118, 2000.
- [28] REZK-SALAMA C., ENGEL K., HADWIGER M., KNISS J., LEFHON A., and WEISKOPF D. Real-Time Volume Graphics. Course 28, ACM SIGGRAPH, 2004.
- [29] ROETTGER S. and ERTL T. A Two-Step Approach for Interactive Pre-Integrated Volume Rendering of Unstructured Grids. In *Proc. of IEEE Symp. on Volume Vis.*, 2003.
- [30] ROETTGER S., GUTHE S., WEISKOPF D., ERTL T., and STRASSER W. Smart Hardware-Accelerated Volume Rendering. In *Proc. of Eurographics/IEEE Symp. on Visualization*, pages 231–238, 2003.
- [31] ROETTGER S., KRAUS M., and ERTL T. Hardware-Accelerated Volume and Isosurface Rendering Based on Cell Projection. In *Proc. of IEEE Visualization*, 2000.
- [32] SABELLA P. A Rendering Algorithm for Visualizing 3D Scalar Fields. In *Proc. of ACM SIGGRAPH*, pages 51–58, 1998.
- [33] WEILER M., WESTERMANN R., HANSEN C., ZIMMERMAN K., and ERTL T. Level-Of-Detail Volume Rendering via 3D Textures. In *Proc. of IEEE Symp. on Volume Vis.*, 2000.
- [34] WEISKOPF D., ENGEL K., and ERTL T. Interactive Clipping Techniques for Texture-Based Volume Visualization and Volume Shading. *IEEE TVCG*, 9(3):298–312, 2003.
- [35] WESTERMAN R. and ERTL T. Efficiently Using Graphics Hardware in Volume Rendering Applications. In *Proc. of ACM SIGGRAPH*, pages 169–177, 1998.
- [36] WILLIAMS P., MAX N., and STEIN C. A High Accuracy Volume Renderer for Unstructured Data. *IEEE TVCG*, 1998.