Seismic Horizon Tracing with Diffusion Tensors

Thomas Höllt* Markus Hadwiger* Laura Fritz* Philipp Muigg† ‡ Helmut Doleisch‡

Abstract

This work presents a new workflow for the interpretation of seismic volume data, as well as a novel approach to interactively tracing seismic horizons. Instead of interpreting the seismic cube slice by slice, in the proposed workflow interpretation is performed on the planes connecting wells that have been drilled. Thereby the additional data provided by the well logs can easily be used during the interpretation process. Instead of manually picking the seismic horizon, we propose an algorithm which uses numerical integration over a vector field computed with diffusion tensors for automatic tracing, based on a user-defined seed point.

Index Terms: Subsurface Exploration, Seismic Volume Data, Seismic Interpretation

1 Introduction

Seismic interpretation as the first step in building a geological model for efficient oil and gas exploration is becoming more and more important with the decreasing availability of fossil fuels. Computer-aided analysis of the subsurface is widely used to reduce drilling costs and obtain detailed information about the size and location of oil and gas reservoirs. The seismic data are acquired by sending sound waves into the Earth by detonating explosives and measuring the resulting reflections. The obtained 3D seismic volumes are then interpreted by geoscientists. This interpretation is often done manually, working from slice to slice in the *inline* or *crossline* directions, i.e., parallel to one of the main axes. With growing data sizes, manual interpretation becomes less feasible, and therefore image processing techniques are used to aid the interpretation process. In addition to the seismic volume, data acquired by drilling wells are available, which results in *well logs*. These are used to support the user during the interpretation procedure. In this work, we propose a novel workflow for seismic interpretation, based on working from well log to well log, instead of in axis-aligned slices, and we present an automated tracing algorithm that exploits diffusion tensors.

2 Related Work

The Seismic Analyzer [5] presented by Patel et al. is an integrated solution for quickly interpreting and illustrating 2D seismic images. The authors present a simple approach to trace seismic horizons and use them to illustratively render the data. Jeong et al. [2] present interactive 3D seismic fault detection utilizing graphics hardware. They use diffusion filtering in a preprocessing step to smooth the volume data and perform the fault detection similarly to edge detection in conventional image processing. Diffusion filtering has been analyzed and presented in detail [7], including the theoretical background and different applications. Our visualization and interpretation solution has been implemented as a plugin for the SimVis framework [1].

*Thomas Höllt, Markus Hadwiger and Laura Fritz are with the VRVis Research Center, Vienna, Austria. E-mail:{hoellt|msh|laura}@vrvis.at

†Philipp Muigg is with the University of Technology, Vienna, Austria

‡Philipp Muigg and Helmut Doleisch are with the SimVis GmbH, Vienna Austria. E-mail:{muigg|doleisch}@simvis.at

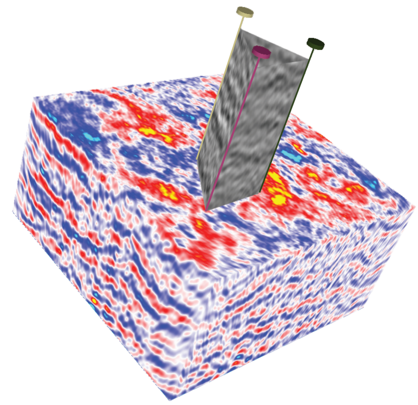

Figure 1: A seismic cube with a three-sided prism for interpretation, resulting from the connection of three user-selected wells.

3 Workflow

The current workflow for seismic interpretation is usually based on manual or semi-automatic interpretation of every single *inline* and *crossline* slice, so that a depth value for a specified seismic horizon is assigned to every (x,y)-coordinate, where the (x,y) plane is parallel to the Earths surface. Even though well logs provide additional information, they require further interpretation and consistent integration with the information contained in the seismic volume.

Furthermore, structures between well logs are hard to identify as neighboring logs usually are not on the same inline or crossline slice.

In collaboration with our industrial partner, we developed a workflow that is based on the position of the wells rather than the axis-aligned inline and crossline slices. The user can define a set of wells, usually three of them, which are then used to define the sides of an n-sided right prism (compare Figure 1). For user interaction, the faces of the prism are unfolded onto a single 2D plane.

Starting from a user-selected seed point, the tracing is then done automatically along the faces of the prism. This process makes it much easier to evaluate the quality of the tracing according to the well logs, compared with the standard tracing methods based on inline and crossline interpretation.

4 Seismic Horizon tracing with Diffusion Tensors

Edge-enhancing anisotropic diffusion [7]is a non-linear smoothing technique which preserves edges by computing an edge direction vector for every sample and using this vector as a parameter for the smoothing operation.

$$\partial u_t = \mathbf{div}(D \cdot \nabla u), \tag{1}$$

where D is the diffusion tensor, and ∇u is the gradient of the intensity volume $u(x,y,z)$. Together, $D \cdot \nabla u$ determines the direction which the divergence operator uses to smooth the image. In the general 3D case, the diffusion tensor is computed as

$$D = [v_1 v_2 v_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \end{bmatrix}. \tag{2}$$

Based on the edge-enhancing diffusion tensor, in our system the first basis vector v_1 is computed by applying a $7 \times 7 \times 7$ Gaussian derivative kernel to the volume at each sample point. This results in

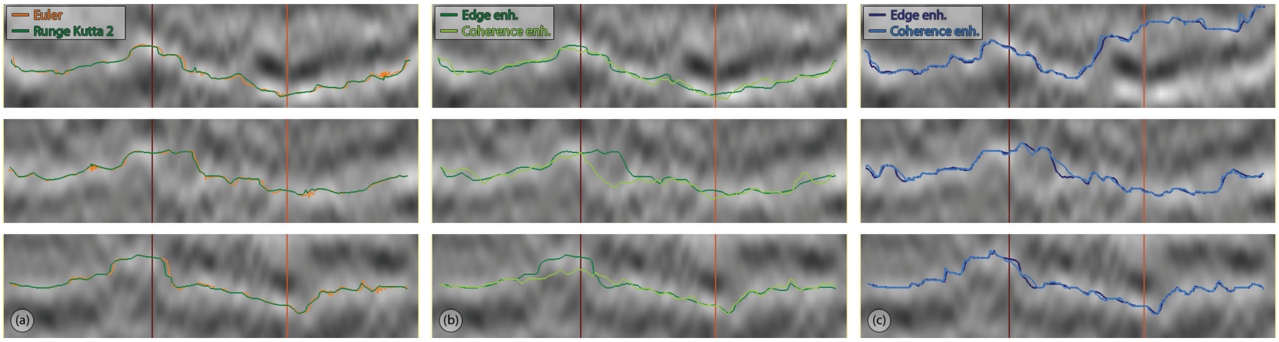

Figure 2: Results of the different seismic horizon tracing methods we have implemented. (a) Euler vs. second-order Runge Kutta Integration, (b) edge-enhancing vs. coherence-enhancing diffusion tensor and (c), just as (b), but with snapping to the largest scalar value.

the smoothed gradient∇u_σ at position (x,y,z). A vector orthogonal to v_1 is then chosen as v_2, and v_3 is a vector orthogonal to both, v_1 and v_2. λ_1, λ_2, and λ_3 are scaling factors, which are responsible for rotating the gradient in the direction of the edge in order to prevent smoothing across it. We define λ_1 as the diffusivity $g(||\nabla u||)$, and $\lambda_2 = \lambda_3 = 1$. To compute the diffusivity, we use the Perona-Malik model [6]:

$$g(||\nabla u||) = e^{-\left(\frac{||\nabla u||}{k}\right)^2}. \tag{3}$$

For our horizon tracing approach, we only need the vector field computed by $D \cdot \nabla u$. However, even though we compute the vector field for each side of the interpretation prism using the whole neighborhood in 3D, the tracing itself is done only on the 2D slices. For this, the vectors need to be projected onto the sides of the prism. The whole procedure of creating the vector field is done in real-time on the GPU with a shader written in the Cg language [3].

The tracing itself is then performed by numerically integrating the vector field from a user-selected seed point. For this integration, we have implemented a second-order Runge-Kutta method in the given vector field as well as simple Euler integration. A comparison can be seen in Figure 2a for a step size of 1.

Jeong et al [2] use coherence-enhancing anisotropic diffusion filtering as basis for fault detection in seismic data. In that work, the tensor is computed as shown in Equation 2. However, instead of directly using the smoothed gradient to create the diffusion tensor, here the Eigenvectors of the structure tensor J at each position are used as v_1, v_2, and v_3, in order of their contribution. J is computed as

$$J_\rho(\nabla I_\sigma) = K_\rho \star (\nabla I_\sigma \otimes \nabla I_\sigma), \tag{4}$$

with

$$K_\rho(x) = \frac{1}{\sqrt{2\pi}\rho} e^{-\frac{x^2}{\rho^2}}, \; I_\sigma = K_\sigma \star I \tag{5}$$

and $\star$ beeing the convolution operator. Additionally $\lambda_1 \approx 0$ and $\lambda_2 = \lambda_3 = e^{-\frac{(v_i \cdot \nabla I)^2}{k^2}}$.

For comparison, we have also implemented a variant of our approach using the coherence-enhancing diffusion tensor. Results showing the tracing, both with the coherence-enhancing, as well as the edge-enhancing diffusion tensor are shown in Figure 2b.

Finally, as the horizons are marked by local extrema in the scalar data, we have implemented an additional version of both approaches that incorporates searching the scalar field for a local maximum or minimum in the tracing algorithm. Rather than using the result of an integration step directly as input for the next step, the neighborhood of a user-defined size in the $-z$ and $+z$ directions is scanned for the highest or lowest scalar value, respectively. The location of this value is then used instead of the integration result as support for the traced horizon, as well as as input for the next integration step. Results for this method can be seen in Figure 2c.

5 Results and Conclusion

Figure 2 shows some exemplary results of the different tracing algorithms. It can be seen that the higher-order Runge-Kutta integration results in much smoother horizons than simple Euler integration (Figure 2a) for all three test traces. Comparing tracing using the edge-enhancing diffusion tensor with the coherence-enhancing tensor (Figure 2b) shows that the former is a little closer to the actual areas of maximum amplitude and more responsive to changes in the horizon direction. The difference between both incorporating the snapping to maximum intensity values is very small (Figure 2c), and for both the correctly traced parts are closest to the actual horizon of all compared methods. However, the deviation from the vector field there introduces a much higher chance of false classification, as can be seen on the right side of the uppermost image.

It can be said that the presented approach is very promising for automatically or semi-automatically tracing seismic horizons. For the immediate future, we plan to use the traced horizons on the faces of the prisms as seeds for building a complete surface mesh inside each prism. The complete pipeline of building the vector fields, tracing the horizons, and creating the surface mesh, will be implemented on the GPU using NVIDIA CUDA [4].

Acknowledgements

We thank Heinemann Oil GmbH for providing the datasets. This work was supported in part by the Austrian Research Funding Agency (FFG) in the scope of the project "SEIVIS" (No. 818063).

References

[1] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the 5th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*, pages 239–248, 2003.

[2] W.-K. Jeong, R. Whitaker, and M. Dobin. Interactive 3d seismic fault detection on the graphics hardware. In *Eurographics / IEEE VGTC Workshop on Volume Graphics*, pages 111–118, July 2006.

[3] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard. Cg: a system for programming graphics hardware in a c-like language. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 896–907, New York, NY, USA, 2003. ACM.

[4] NVIDIA Corporation. *NVIDIA CUDA Compute Unied Device Architecture Programming Guide Version 1.1*, NVIDIA Corporation, 2007.

[5] D. Patel, C. Giertsen, J. Thurmond, J. Gjelberg, and M. E. Gröller. The seismic analyzer - interpreting and illustrating 2d seismic data. *IEEE Transaction on Visualization and Computer Graphics*, 14(6):1571–1578, Oct. 2008.

[6] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[7] J. Weickert. *Anisotropic Diffusion In Image Processing*. PhD thesis, University of Kaiserslautern, 1996.