Tu LHR1 15

High Performance GPGPU Structure-preserving Smoothing for Seismic Amplitude Data by Anisotropic Diffusion

G.M. Faustino* (Tecgraf/PUC-Rio), P.C. Pampanelli (Tecgraf/PUC-Rio), J.M. V. Duarte Junior (Tecgraf/PUC-Rio), E.A. Perez (Tecgraf/PUC-Rio), E.R. Silva (Tecgraf/PUC-Rio), P. Frederick (Tecgraf/PUC-Rio) & P.M.C. Silva (Tecgraf/PUC-Rio)

SUMMARY

Noise attenuation plays an important role in seismic data processing and interpretation. In recent years, the anisotropic diffusion filter has received much attention since it has superior performance in edge-preserving while smoothing noise from noisy signals. This work presents a high-performance General Purpose Graphics Processing Unit (GPGPU) structure-preserving smoothing for seismic amplitude data by anisotropic diffusion. The obtained results show that the proposed method runs in iterative time, and is able to remove noise and preserves structural features efficiently. We also compare the computational performance of CPU and GPU implementations and show that GPU is about 14 times faster.

Introduction

Seismic data are usually contaminated with noise. It can appear as a result of many different physical processes and, consequently, with many different seismic characteristics. Thus, removal of the noise is a key component in preparing the data for all others signal processes to work more effectively. In addition, seismic attributes have been used to accelerate interpretations, and their results are directly related to the quality of the seismic data. Therefore, adequate and proper noise attenuation methods, i.e., those that remove noise while preserving relevant details such as structural and stratigraphic discontinuities, help to maximize the potential benefit and contribution of the seismic data in exploration.

Although, effective noise removal is critically important, the implementation of those methods can be very computing-intensive. For a huge amount of input data, the time spent filtering the data can be prohibitive. Also, a continuous increasing of size in three-dimensional seismic data has been noticed in the past decades. Currently, in order to filter these huge data, a large amount of time may be required, even in a high-performance workstation. Hence, the search for new technologies capable of accelerating processing speed, and consequently reduce processing time can leverage the seismic interpretation process.

The anisotropic diffusion filter has received much attention since it was first proposed by Perona and Malik (1990) in the nineties. Unlike conventional spatial filters that blur boundaries or edge structures, anisotropic diffusion filter can eliminate noise and preserve, or even enhance edges simultaneously. A large number of approaches using diffusion techniques have been proposed, such as Hocker and Fehmers (2002), Hale (2009), Baddari et al. (2011) and Zhang et al. (2015). However, although they provide good results, its is important to mention that these works use the seismic amplitude gradient to orient the diffusion process. According to Silva et al. (2012), the seismic amplitude gradient is not appropriate since it naturally presents variations along the horizons. In addition, such variations can make the diffusion occur in the perpendicular direction to the horizon, which is not desirable. Differently from our method, the work described in Zhang et al. (2015) apply the anisotropic diffusion process for filtering the coefficients of the Shearlet Transform, not the seismic amplitude data.

In this work, we present a high-performance General Purpose Graphics Processing Unit (GPGPU) structure-preserving smoothing for seismic amplitude data by anisotropic diffusion. Our formulation of the diffusion process uses seismic attributes to identify horizons and faults, which are used as constraints of the proposed method. Differently from previous works, we use the instantaneous phase gradient to orient the diffusion process, and an implicit method for solving the diffusion equation. For each filtering step, a system of equations must be resolved. Therefore, given the huge amount of input data, usually dozens of GB, the use of GPGPU techniques for solving the numerical system are essential. In order to evaluate the proposed method, experimental tests were done on real seismic data. The results show that this filter is effective in removing noise, while preserving structural features. The authors also compare the computational performance of CPU and GPU, and show that GPU is about 14 times faster. We achieve a speed time of 56 milliseconds for a seismic section of size 951×462 samples. With this processing time, we are able to smooth the desired seismic data on demand during the interpretation process.

Structure-preserving Smoothing Method

The method implemented in this paper is based on two principal steps. At first, the attributes that identify horizons and faults are calculated. Here, the seismic reflectors are properly described as level surface, through the computation of a Horizon Identifier Attribute (Silva et al., 2012). In order to estimate the discontinuities (faults), we use the attribute described in Pampanelli et al. (2014). In this article, the authors presented a fault attribute based on the first directional derivative of the complex trace.

On the second step, we have to assemble and solve the system of equations. In other words, filter the seismic amplitude input data. The horizons and faults attributes presented previously represent the constraints of the diffusion process. The amplitude gradient represents the flux from the regions of higher concentration to regions of lower concentration. The diffusion equation, $\frac{\partial u}{\partial t} = \nabla \cdot (\varepsilon D \vec{\nabla} u)$, is a function of the amplitude u varying in time t, where $\nabla\cdot$ is the divergence operator, and D indicates the diffusion tensor. The coefficient ε represents the so-called fault preserving factor. Taking into account the direction given by the instantaneous phase gradient, we calculate the diffusion tensor $D = \vec{\nabla}\Phi^{\perp} \times (\vec{\nabla}\Phi^{\perp})^T$, where $(\vec{\nabla}\Phi^{\perp})^T$ means the transposed vector of $\vec{\nabla}\Phi^{\perp}$. As a result, $[D]$ is a symmetric positive semidefinite matrix. Finally, the direction of the diffusion process is defined by this tensor. The final equation is presented below:

$$u_{x,y}^{n+1} = u_{x,y}^{n} + \Delta t \nabla(\varepsilon_{x,y}^{n} D_{x,y}^{n} \nabla u_{x,y}^{n+1}), \tag{1}$$

where Δt represents the time increment of the diffusion process.

The Equation 1 is applied for each sample of the input seismic data. As a result, the last step of our method consists in solve a system of equations that can be written as a matrix equation in the form $[A]\vec{x} = \vec{b}$. For each filtering step, the solution of the system is the new seismic amplitude filtered by the anisotropic diffusion. Since this is an iterative method, the anisotropic diffusion filtering can be applied as many times as needed until the desired amplitude data is obtained.

Parallel Approach

In this section, we are going to describe the strategy used to parallelize the structured filtering method. On the first step, we need to calculate the instantaneous phase gradient $(\vec{\nabla}\Phi)$ that involves computing the Hilbert transform. We use Fast Fourier Transform in parallel on each trace to convert between domains before applying the Hilbert transform. We used FFTW3 and CUFFT libraries to support the implementation of CPU and GPGPU code, respectively.

On the second step, since the Equation 1 is applied for each sample, filtering the seismic section consists in solving a linear system of size $m \times m$, where $m = w \times h$ and w and h represents the width and height of the input data, respectively. The derivative operators are discretized using central differences with three samples. As a consequence, for each sample a nine-point stencil is used. Therefore, considering the linear system in the matricial form $[A]\vec{x} = \vec{b}$, the matrix A is sparse, $\vec{x}$ is our filtered data and $\vec{b}$ is the input data.

Many algorithms can be used for solving this kind of system. Krylov subspace methods are well suited for large and sparse linear systems, and in this work we choose to evaluate the following numerical methods: Conjugate gradient (CG), Biconjugate gradient stabilized (BiCGStab), Biconjugate gradient (BiCG) and Generalized minimal residual (GMRES). The CG method is designed to solve systems of linear equations whose matrix is symmetric and positive-definite. In order to use CG method, we have to multiply the system equation by A^T to guarantee that it will be symmetric and positive-definite. The equation for CG method will be $[A^T][A]\vec{x} = [A^T]\vec{b}$. The other solvers can be used with nonsymmetric linear systems, and do not require changes on the original equation.

Regardless the numerical method used to solve the system, each seismic section is filtered one at a time. Since this is an iterative method, the anisotropic diffusion filtering can be applied as many times as needed until the desired amplitude data is obtained. However, if necessary, due to memory restrictions the image may be split into subimages.

Results

In this section, we present some results and compare the computational performance of CPU and GPU with regard to the diffusion filtering method described in this abstract. The experiments were done on a Z820 Workstation equipped with 64GB RAM, an Intel(R) Xeon(R) E5-2620 v2 @ 2.10 GHz CPU, and a Tesla K40 12GB GPU Accelerator with Kepler Micro-architecture, and 2880 Thread Processors.

We performed the filtering experiment on the volume of the Netherlands offshore F3 block downloaded from the Opendtect Repository (OpenSeismicRepository, 2015). Aiming to evaluate the numerical methods presented in the previous section, we use the Intel Math Kernel Library 11.3 (MKL) (Intel, 2007) for CPU implementation, and NVIDIA CUSP 0.5.1 (Bell and Garland, 2012), and ViennaCL 1.7.0 (Rupp et al., 2010) libraries for GPU implementation.

In order to evaluate each numerical method, we ran the described filter 12 times. We discarded the highest and lower execution time, and then take the average time. In each scenario, we choose one numerical method, one library and the number of iterations. Table 1 shows the execution time in milliseconds for each scenario. For these tests, we used the inline section 240 of F3 block seismic data, which has a size of 951 $\times$ 462 samples. The stop criteria for all scenario were a tolerance error of 10^{-4} and a limit of 100 iterations.

Processing Speed Comparing the CPU and GPU Implementations

Number of Iterations	*CG*			*GMRES*			*BICG*	*BICGSTAB*	
	MKL (ms)	***CUSP (ms)***	*ViennaCL (ms)*	*MKL (ms)*	***CUSP (ms)***	*ViennaCL (ms)*	*CUSP (ms)*	***CUSP (ms)***	*ViennaCL (ms)*
1	524	**64**	65	379	**56**	70	76	**61**	66
2	1009	**95**	115	736	**80**	124	120	**91**	117
3	1549	**124**	164	1093	**102**	179	162	**120**	168
4	2028	**154**	213	1453	**123**	232	204	**146**	217
5	2523	**183**	262	1800	**145**	284	245	**172**	268
6	3013	**212**	311	2159	**167**	337	287	**199**	318
7	3510	**241**	360	2519	**189**	389	329	**225**	367
8	3976	**268**	409	2858	**211**	442	370	**252**	417
9	4429	**298**	458	3231	**233**	495	412	**278**	467
10	4891	**325**	507	3574	**253**	547	454	**304**	517

Table 1 *Executions time in milliseconds of the numerical methods for each library. The diffusion filter was applied on a seismic section of size 951 $\times$ 462 samples. It is important say that the MLK library provides the FGMRES method instead of GMRES. The columns in bold represent the best execution time for each numerical method.*

As shown in Table 1, the NVIDIA CUSP library achieved the best results with a lower execution time for all numerical methods, varying the number of iterations. The GMRES with CUSP presented the fastest results. Figure 1 depicts the speed up between MKL (CPU) and CUSP (GPU) libraries with GMRES method. As we can see, the GPU is up to 14 times faster than CPU. Figure 2 shows the inline section 240 of F3 block seismic data and the corresponding filtered image. As can be seen, the structural features are preserved and even enhanced. Therefore, with this results, we have a structure-preserving noise attenuation method that runs in iterative time.

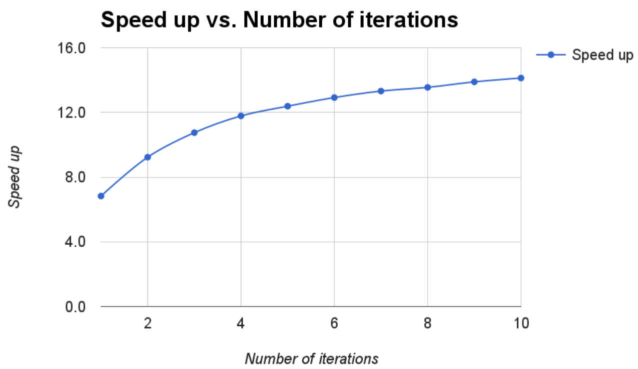

Figure 1 *Plots of GMRES speed up versus number of iterations.*

Conclusions

The recognizing and the tracking of seismic events in the presence of noise is especially important for processing and further interpretation. Seismic attributes have been used to accelerate interpretations, and their results are directly related to the quality of the seismic data. In this abstract, we presented a structure-preserving filtering method based on the anisotropic diffusion of the amplitude field that runs in iterative time. The obtained results show that the proposed method is able to remove noise and preserves structural features faster and efficiently. As a consequence, it can be used to improve the response of seismic attributes on demand during the interpretation process. We also compared the computational performance of CPU and GPU, concluding that GPU is about 14 times faster. We observed that the implementation using GMRES from CUSP achieved the fastest results.

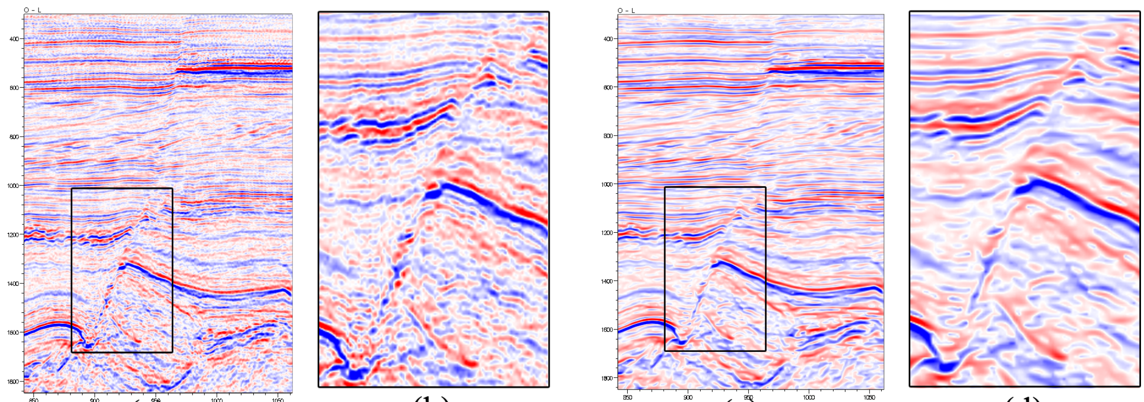

Figure 2 *(a) Inline section 240 of F3 block seismic data; (b) zoom area of the input data; (c) and (d) corresponding results for 3 iterations.*

Acknowledgments

The authors would like to thank the Computational Geophysics Group from Tecgraf/PUC-Rio.

References

Baddari, K., Ferahtia, J., Aïfa, T. and Djarfour, N. [2011] Seismic Noise Attenuation by Means of an Anisotropic Non-linear Diffusion Filter. *Comput. Geosci.*, **37**(4), 456–463.

Bell, N. and Garland, M. [2012] Cusp: Generic parallel algorithms for sparse matrix and graph computations. *Version 0.3. 0*, 35.

Hale, D. [2009] Structure-oriented smoothing and semblance. Tech. rep., Center for Wave Phenomena, Colorado School of Mines, Golden CO, USA.

Hocker, C.F.W. and Fehmers, G.C. [2002] Fast structural interpretation with structure-oriented filtering. *The Leading Edge*, **21**(3), 238–243.

Intel, M. [2007] Intel math kernel library.

OpenSeismicRepository [2015, https://opendtect.org/osr/] Open Seismic Repository.

Pampanelli, P., Faustino, G., Silva, P.M., Kolisnyk, A. and Gattass, M. [2014] A New Fault-Enhancement Attribute Based on First Order Directional Derivatives of Complex Trace. In: *EAGE*.

Perona, P. and Malik, J. [1990] Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, **12**(7), 629–639.

Rupp, K., Rudolf, F. and Weinbub, J. [2010] ViennaCL-a high level linear algebra library for GPUs and multi-core CPUs. *Proc. GPUScA*, 51–56.

Silva, P.M., de Oliveira Martins, L. and Gattass, M. [2012] *Horizon Indicator Attributes and Applications*, chap. 227. Society of Exploration Geophysicists, 1–6.

Zhang, C., Li, Y., Lin, H., Yang, B. and Wu, N. [2015] Seismic Random Noise Attenuation and Signal-preserving by Shearlet Transform Based on Anisotropic Diffusion. In: *EAGE 2015*.