

Flow-based Segmentation of Seismic Data

Kari Ringdal*

Supervised by: Daniel Patel†

Institute of Informatics
University of Bergen
Bergen / Norway

Abstract

This paper presents an image processing method for identifying separate layers in seismic 3D reflection volumes. This is done by applying techniques from flow visualization and using GPU acceleration. Sound waves are used for exploring the earth beneath the surface. The resulting seismic data gives us a representation of sedimentary rocks. Analysing sedimentary rocks and their layering can reveal major historical events, such as earth crust movement, climate change or evolutionary change. Sedimentary rocks are also important as a source of natural resources like coal, fossil fuels, drinking water and ores. The first step in analysing seismic reflection data is to find the borders between sedimentary units that originated at different times. This paper presents a technique for detecting separating borders in 3D seismic data. Layers belonging to different units can not always be detected on a local scale. Our presented technique avoids the shortcoming of existing methods working on a local scale by addressing the data globally. We utilize methods from the fields of flow visualization and image processing. We describe a border detection algorithm, as well as a general programming pipeline for preprocessing the data on the graphics card. Our GPU processing supports fast filtering of the data and a real-time update of the viewed volume slice when parameters are adjusted.

Keywords: Seismic Data, Structure Extraction, GPU-accelerated Image Processing

1 Introduction

Stratigraphy is the study of rock layers deposited in the earth. A stratum (plural: strata) can be defined as a homogeneous bed of sedimentary rock. Stratigraphy has been a geological discipline ever since the 17th century, and was pioneered by the Danish geologist Nicholas Steno (1638-1686). He reasoned that rock strata were formed when particles in a fluid, such as water, fell to the bottom [10]. The particles would settle evenly in horizontal layers on a lake or ocean floor. Through all of Earth's history, layers

of sedimentary rock have been formed as wind, water or ice has deposited organic and mineral matter into a body of water. The matter has sunk to the bottom and consolidated into rock by pressure and heat. A break in the continuous deposit results in an *unconformity*, in other words, the surface where successive layers of sediments from different times meet. An unconformity represents a gap in the ge-



Figure 1: Stratified sediments. The sedimentary facies are separated by unconformities.

ological record. It usually occurs as a response to change in the water or sea level. Lower water levels expose strata to erosion, and a rise in the water level may cause new horizontal layers of deposition to resurge on top of the older truncated layers. The geologists analyse the pattern around an unconformity to decode the missing time it represents. Above and below an unconformity there are two

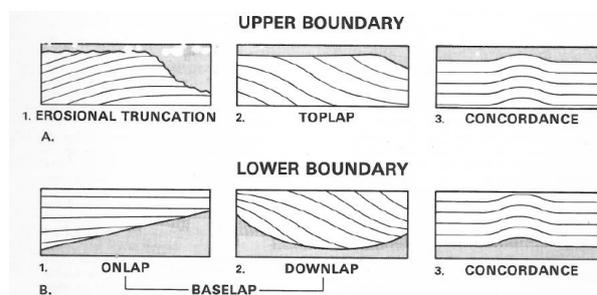


Figure 2: Strata relates to an unconformity in different ways. The top three images show patterns that occur below an unconformity, and the bottom three images show patterns that occur above an unconformity.

types of terminating patterns and one non-terminating pat-

*kari.ringdal@student.uib.no

†danielpatel.no@gmail.com

tern. Truncation and top lap terminates at the unconformity above. Truncation is mainly a result of erosion and top lap is a result of non-deposition. Strata above an unconformity may terminate in the pattern of onlap or downlap. Onlap happens when the horizontal strata terminates against a base with greater inclination, and downlap is seen where younger non-horizontal layers terminates against the unconformity below. Concordance can occur both above and below an unconformity and is where the strata layers are parallel to the unconformity. An illustration of these concepts can be seen in Figure 2.

An unconformity can be traced into its *correlative conformity*. In contrast to an unconformity, there is no evidence of erosion or non-deposition along the conformity.

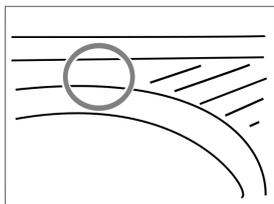


Figure 3: Sediments of different facies can be indistinguishable in local areas such as inside the circle.

A seismic sequence - also called a sedimentary unit or facies, is delimited by unconformities and their correlative conformities. The fact that sediments belonging to different seismic sequences can be indistinguishable in greater parts of the picture, as illustrated in Figure 3, calls for global analysing tools.

For more information on unconformities, sedimentary sequences and other stratigraphic concepts, the reader is

referred to Nichols book on sedimentology and stratigraphy [13] and Catuneaus book on sequence stratigraphy [4].

Many techniques to highlight interesting attributes in seismic data have been developed [6]. Taner [17] gives a useful definition of seismic attributes: "Seismic Attributes are all the information obtained from seismic data, either by direct measurement or by logical or experience based reasoning". *Dip* and *azimuth* are attributes that describe the dominating orientation of the strata locally. Dip gives the vertical angle and azimuth the lateral angle.

In our work, we consider the dip/azimuth vectors to constitute a flow field representation of the data set where the flow "moves" along the sediment layers. We seed particles from neighbouring points in this flow, and consider the distances between the end points of the generated trajectories. A great distance gives a high probability that the seed point is a surface point. Mapped surface points are then linked to constitute unconformity surfaces. Figure 4 gives an overview of our processing pipeline.

2 Related Work

Interpreting seismic data is a time consuming task, and extensive work has been done to automate this. This section will focus on previous approaches on finding unconformities, and also look into methods within the fields of image processing and flow visualization that relates to the

new technique presented in this paper. Orientation field extraction from image processing relates to vector field extraction of seismic data. Image processing also deals with edge linking methods, which relates to the segmentation process of our method. The field of flow visualization use methods relevant to the mapping of surface probability.

2.1 Seismic methods

One method for detecting sequence boundaries, or unconformities, is the method of Randen et al. [15]. This method calculates the volumetric estimates of dip and azimuth by applying a multi-dimensional derivative of a Gaussian filter followed by directional smoothing. Starting at a sample in the extracted orientation field, a curve is generated in the direction of the local angle (see Figure 5).

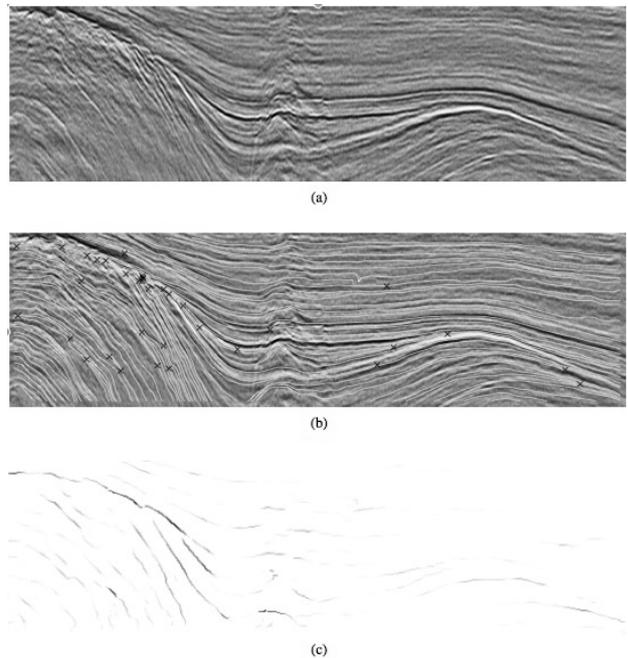


Figure 5: a) Seismic cross section. b) Flow lines and terminations (marked with X) extracted from the cross section. c) Mapping of stratigraphic surface probability from the cross section.

The curves form flow lines along the orientation field. Intersection points of flow lines are detected (marked with X). These points are likely to be on an unconformity. Brown and Clapp attempted a different approach that locally compares the data to a template that represents a neighbourhood around an unconformity [2]. Another method to find lateral discontinuities (e.g. lateral unconformities and faults) in seismic data is that of Bahorich and Farmer called the coherence cube [1]. Coherency coefficients are calculated from a 3D reflection volume and displayed as a new volume. Coherence is a measure of the lateral change of the seismic waveform along structural dip. Since the coherence cube first appeared in 1995 it has been

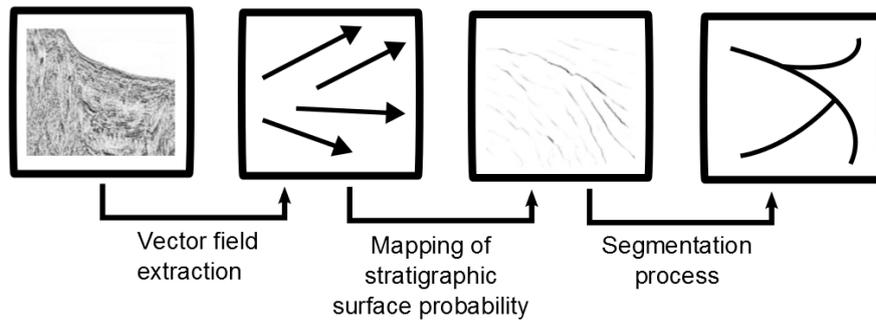


Figure 4: Processing pipeline. A flow field with vectors parallel to the sediment layers is extracted from the data. Stratigraphic surface probability is mapped by the use of a seeding algorithm and user defined parameters. Edge linking can be a way of completing a segmentation.

improved several times. Chopra gives an overview of the development of this method [5]. Unconformities are often seen as discontinuities in the data, but not always. It can happen that there are no obvious signs of erosion and the layers on either side of an unconformity are parallel. This type of unconformity, sometimes called paraconformity or conformity, would not be detected by the coherence cube or any of the above methods.

A recent paper by Hoek et al. [18] describes a new method for finding unconformities. Gaussian derivative filters are used to estimate the dip/azimuth field. The orientation field is then analysed by utilizing a method from the field of image processing. The structure tensor field is calculated and regularized, and the principal eigenvector of the structure tensor is extracted. From this, the dip field is studied to see whether the vectors diverge, converge or are parallel. Hoek et al. recognize the problem that previous methods address seismic data on a local scale, and they attempt to find a more global approach with their unconformity attribute. However, their method measures the conformability of the dip field in a neighbourhood of a predefined size and is therefore still a local method that does not capture events taking place outside its neighbourhood.

Hoek et al.'s unconformity detection method, as well as the method presented in this paper, depends on the reflector dip and azimuth of the seismic data. Much work has been done to extract these attributes accurately. Complex trace analysis, discrete vector dip scan, and gradient structure tensor are commonly used for the task. Marfurt presents a refined method for estimating reflector dip and azimuth in 3D seismic data and gives a good overview of the work previously done in this area [12].

2.2 Image processing methods

In image processing, a repetitive pattern is referred to as a texture, and a linear pattern as an oriented texture. Numerous algorithms are used for enhancing or segmenting textured images - many inspired by human visual perception models. When it comes to processing images digitally for tasks such as edge detection or pattern recognition, there

is no algorithm generic enough to be a good choice at all times. It is in other words necessary to choose the right algorithm for the right data and desired achievement.

A finger print, wood grain or a seismic image are examples of oriented textures. These textures have a dominant orientation in every location of the image. They are anisotropic, and each point in the image has a degree of anisotropy that relates to the rate of change in the neighbourhood. This is often represented as the magnitude of the orientation vectors. Extraction of the orientation field of a texture has been well researched in the field of image processing. Extraction algorithms are often based on the gradient from a Gaussian filter [16, 9]. In addition to automatic pattern recognition and some edge detection algorithms, directional smoothing of an image exploits the orientation field. Like in seismic data evaluation, it is essential that the extracted orientation field represents the intrinsic properties of the image.

Non photo-realistic rendering (NPR) concerns with simplifying visual cues of an image to communicate certain aspects more effectively. Kang et al. [8] suggests a new NPR method for 2D images that uses a flow-based filtering framework. An anisotropic kernel that describes the flow of salient image features is employed. They presents two topics that are interesting to our technique, namely the extraction of a vector field from an image, and creating lines from isolated points. They use a bilateral filter (an edge-preserving smoothing filter) for the construction of what they call an edge tangent field (ETF). This is a vector field perpendicular to the image gradients. The gradient map is obtained by a Sobel operator. The vector adapted bilateral filter takes care to preserve salient edge directions, and to preserve sharp corners in the input image. The filter may be iteratively applied to smooth the ETF without changing the gradient magnitude. Kang et al. present this vector field extraction method as a base for extracting image features. A different vector field extraction method is proposed by Ma and Manjunath [11]. They find edge flow vectors by identifying and integrating the direction of change in color, texture and phase discontinuity at each image location. The vector points in the direction of

the closest boundary pixel.

Edge linking is another area from image processing that is relevant to our method. An image of unconformity lines may contain gaps, and with an ultimate goal of segmentation, proper linking of edges is necessary. Fundamental approaches to edge linking concern both local processing where knowledge about edge points in a local region is required, and regional processing where points on the boundary of a region must be known [7]. There are also global processing methods, like the Hough transform. For Kang et al.'s flow-based image abstraction method [8], part of the goal is to end up with an image-guided 2D line drawing. Here the lines are generated by steering a DoG (difference of Gaussian) edge detection filter along the ETF flow and accumulate the information. This way, the quality of lines is enhanced. (see Figure 6).

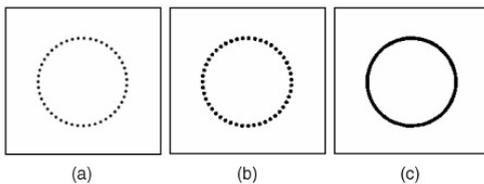


Figure 6: Edge linking. a) Input image. b) Filtered by DoG filter. c) Filtered by Kang et al.'s flow-based DoG filter.

2.3 Flow field topology and extraction methods

Flow visualization is a sub-field of data visualization that develops methods to make flow patterns in fluids visible. Flow features and techniques for topology extraction of steady vector field data will be the focus of this section. A feature is a structure or an object of interest. Shock waves, vortices, boundary layers, recirculation zones, and attachment and separation lines are examples of flow features.

Relating flow to seismic data, features that are most likely to occur in a vector field extracted from seismic data are separation and attachment lines, this because of the onlap, toplap and downlap terminations. Separation and attraction lines are lines on the boundary of a body of a flow where the flow abruptly moves away from or returns to the surface of the flow body. A state of the art report by Post et al. [14] deals with different methods for separation and attachment line extraction. Methods for both open and closed separation are discussed. One approach mentioned is particle seeding and computation of integral curves. A particle is released into the flow field and its path is found by integrating the vector field (that represents the flow field) along a curve. If we look at the vector field extracted from seismic data as a flow field, we have a steady flow. The fact that it is not time-dependent means that the pathline of a seeded particle is everywhere tangent to the vectors of the flow. According to the aim of this pa-

per, feature extraction and its instrumental algorithms are of greater interest than the actual visualization of the data. We will not use pathlines for visualization purposes, but as a tool in addressing the seismic data on a global scale.

3 Implementation Details

Our method for separating the sedimentary units in 3D seismic data follows the processing pipeline shown in Figure 4. The processing steps are separated into three categories:

- Vector field extraction from the seismic data
- Mapping of stratigraphic surface probability
- Segmentation process

This section will take a closer look at each of the steps, but focus on the second step of the pipeline - the mapping of stratigraphic surface probability using concepts from flow. In this step lies the novelty of our method.

3.1 Preprocessing - vector field extraction on the GPU

As described in the Related work section, there already exists efficient methods for estimating the reflector dip and azimuth of seismic data. The extraction of this vector field is an important step of the pipeline, because a regular vector field, that represents the data accurately, is crucial to our technique. The idea is to apply our unconformity extraction algorithm on a dip/azimuth vector field found by already established methods. For testing the algorithm we have created 2D and 3D synthetic data sets and implemented image processing methods for extracting the vector field. The 3D data sets were created either by stacking an image to form a volume or by procedurally creating a volume of vectors pointing in different directions on either side of a delimiting surface. Only the last mentioned type of test set has a variation in z-direction. The test sets have been useful for testing our algorithm and for getting a feel of the vector field extraction calculations which were done in parallel on a graphics card. The implemented programming pipeline is transferable to real seismic data volumes.

The highly parallel structure of modern GPUs is ideal for efficiently processing large blocks of data provided the calculations could be done in parallel. GPUs support programmable shaders that manipulate geometric vertices and fragments and output a color for every pixel on the screen. Instead of output to the screen, the RGBA-vectors can be written to 2D or 3D textures. A texture is in this sense a block of memory located on the GPU where every point of the texture is a four dimensional vector. GPU-accelerated methods are rapidly expanding within the oil and gas industry, and have dramatically increased the computing performance on seismic data.

For our implementation, we have used the OpenGL API and the OpenGL shading language GLSL within the framework of Volumeshop [3]. The programmed pipeline does GPU filtering of 3D data iteratively with a real-time update of the viewed 2D slice when adding, removing or adjusting any of the filters. This allows a fine-tuning of parameters before the entire data volume is processed.

The volume is loaded on the GPU in a 3D texture. GPU-memory is reserved for two more textures of equal size as the volume, and the original texture is copied to one of them. The two textures are used alternately for reading and writing in a ping-pong fashion while the desired number of filters is applied. The different filters are written as shaders in a plugin. Any number of this plugin is added to the Volumeshop interface. They all operate on the GPU by having one plugin for every filter applied to the volume. The user chooses a filter from a pull down menu, and the

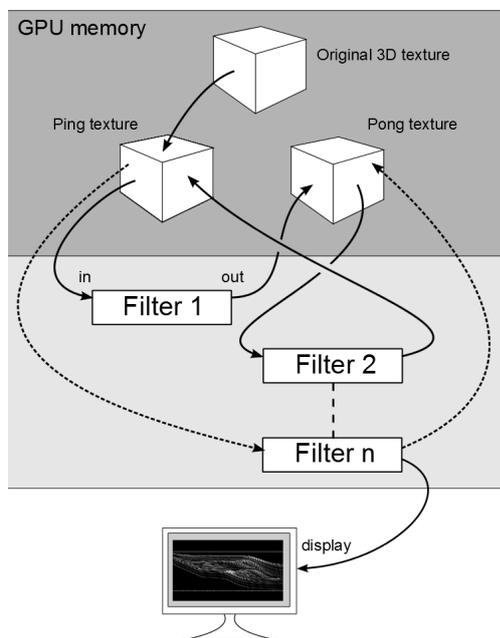


Figure 7: The data is loaded onto the GPU memory in a 3D texture. Data is alternately read and written between two more textures in a ping-pong fashion while filters are applied iteratively. The data is filtered in parallel on the GPU, and a flow field representing the data is extracted. Output from a filter is rendered to the ping or pong textures. Any output slice can also be rendered to the display.

filter parameters can be adjusted by sliders. It is also possible to choose how many slices of the volume are filtered at each step. This way 3D filtering can be done on a subset of the volume to quickly assess the result. Adjustments of a filter leads to reprocessing of the data from the original volume through all the added filters. Therefore, the original volume is kept as a separate texture on the GPU and not overwritten as the ping-pong textures. Figure 7

illustrates the implementation. All calculations are saved with floating point precision in the RGBA-vectors of the 3D textures. Results can be visualised directly by rendering the RGBA-vectors to the screen, i.e. a flow field is seen as colors that varies according to the dip/azimuth vectors. This gives a good indication of the effect of each applied filter. A flow field represented as colors can be seen in Colour plate Figure 12 b).

3.2 Mapping of stratigraphic surface probability

Our unconformity detection algorithm is implemented as a shader, and calculations are done in parallel on the GPU. The technique uses particle seeding, as is common in the field of flow visualization, but the paths of the particles are not visualized. We use particle seeding from four neighbouring points to check whether they belong to the same sedimentary unit or not. The algorithm is as follows: For every point on a volume slice, four seed points are chosen. The seed points have coordinates (x, y, z) , $(x + d, y, z)$, $(x, y + d, z)$ and $(x, y, z + d)$. d is set so all four points are within a local neighbourhood. The seeding is calculated by sampling the flow from the 3D texture using the Runge Kutta 4th order (RK4) method. All four paths are followed until a user-defined number of steps are taken, or the path reaches coordinates outside of the volume-texture. The distances between the four end points are calculated.

If a distance greater than a user-defined threshold is detected, the original seed point with coordinates (x, y, z) is marked as a probable point on an unconformity. Figure 8 is an illustration of the algorithm in 2D. It is expected that a great distance indicate that the particles have moved along differently shaped paths. Because of the parallel nature of the seismic data, two close paths will end up in the same neighbourhood if their seed points are within the same sedimentary unit. We use four seed points to detect borders of any angle. The advantage of this method is that paths of many steps address the data globally and unconformities can be mapped even with parallel horizons on both sides. A premise is that the particles move into a non-parallel area along their paths.

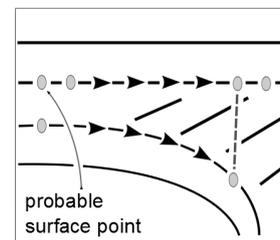


Figure 8: Illustration of the border detection algorithm. The dots represents three seeds that move along paths in a flow field. If the distance between the path-ends exceeds a threshold, a probable surface point is marked.

Since the algorithm is implemented as a shader, it is executed in parallel for every pixel of the rendered region. The rectangular viewing region is set to correspond to the

width and height of the volume, and this region is rendered to the screen or directly to the ping-pong textures. The user can adjust the number of path steps and the distance threshold used by the algorithm. The effect of adjusting these parameters is seen in real-time on one slice of the volume. When the desired parameter values are set, the rest of the volume is processed, slice-by-slice, with the same settings. This way, the seeding algorithm is executed for every vertex in the volume.

When the flow field is sampled from the 3D texture during the path calculations, OpenGL takes care of any necessary interpolation. The textures are set up for trilinear interpolation, and the paths are found by the RK4 integration method. If the step size of this method is set to h , the error per step is on the order of h^5 , while the total accumulated error has order h^4 . Because of the error accumulation, a small step size is desirable. The step size together with the number of steps affect whether the implemented technique is run locally or globally. Since the RK4 calculations are a bottleneck in our technique, the balancing between the accuracy of the method and the performance speed lies in the choice of these parameters. We are using a step size of 0.5. The number of steps is chosen in the user interface.

Ideally, the mapped points constitute unconformity lines without gaps when the seeding is done for every point on a slice, and unconformity surfaces without gaps when the seeding is done on the entire volume. However, this is rare, when dealing with real seismic data, and some fragmentation and false positives may occur. We have taken some measures to reduce misclassification. For more robustness of the algorithm we perform the seeding in both directions of the flow. We also scale the endpoint distances by the number of path steps to make sure that a path of few steps will be as sensitive to the given threshold as a path of many steps. To avoid false positives, compared particle paths are discarded if their paths have a great variation in the executed number of steps.

3.3 The segmentation process

The ultimate goal is a fully automated segmentation of seismic data into sedimentary units. However, when running the algorithm on noisy data, some fragmentation of the detected borders may occur. In that case, edge linking is required. This step is not implemented at the current stage, as it is not the main focus of this work. See the end of Section 2.2 for possible methods to achieve edge linking.

4 Results / Application

To test the idea behind the presented technique, 2D and 3D test sets were generated. The first 2D test set of size 256×256 is a vector field that simulates the flow field of two sedimentary units with parallel layers in greater parts of the picture. As expected, the unconformity surface was

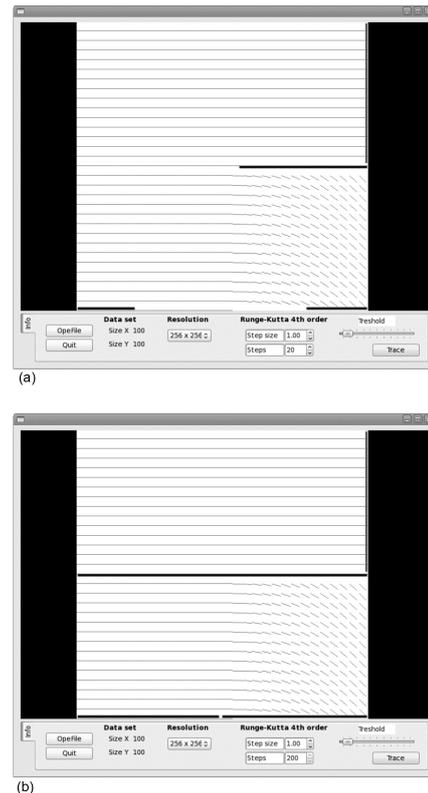
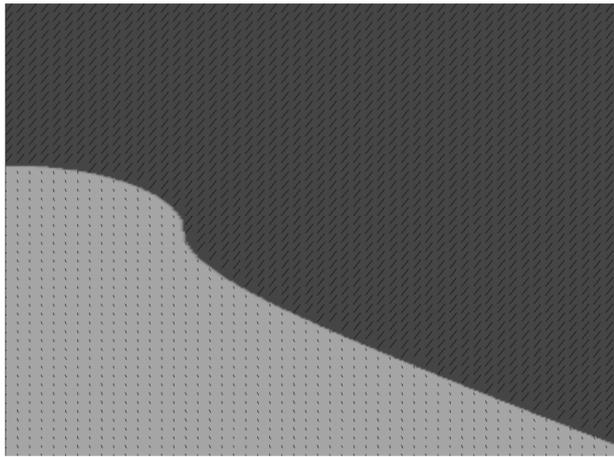


Figure 9: 2D test set of size 256×256 representing the vector field from two sedimentary units. a) The border detection algorithm is run locally - each particle is followed for 20 steps, and only a part of the border is detected. b) The algorithm is run globally - each particle is followed for 200 steps, and the border is detected throughout the data set.

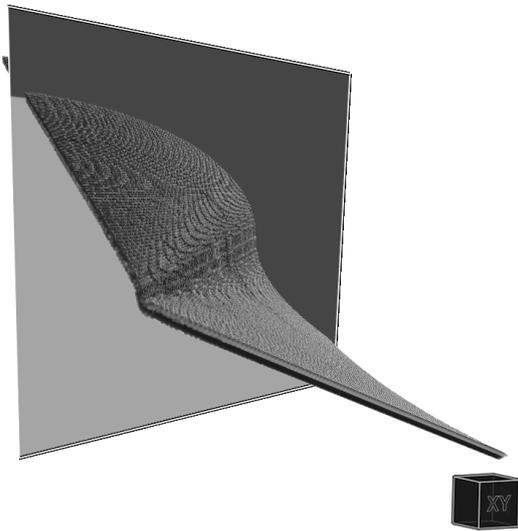
only found in the area without parallel layers when the algorithm was run locally (see Figure 9a). Each particle path had 20 steps with a step size of 1 which means that every particle travelled within a local neighbourhood. Figure 9b shows the result of increasing the number of steps to 200. Now the unconformity was mapped all the way, also in the area of parallel layers.

The second test set is a 3D vector field of size 256×256 . It also simulates two sedimentary units. Figure 10 a) shows the flow field on a slice of the volume. The two units have vectors with different z-components and the data set varies in depth. The seeding algorithm was run on this flow field with 100 path steps and a step size of 0.5, and the unconformity surface was found. Figure 10b shows the detected surface displayed in VoulmeShop. On this simple test set, the algorithm has generated a complete surface, and edge linking is not required.

For test set number 3 we used an image of a seismic data slice from Randen et al.'s article [15], and stacked copies of this image to create a 3D volume. The resulting volume is of size $256 \times 256 \times 256$. This data set does not vary in depth as a seismic volume, but it gives us a



(a)



(b)

Figure 10: Test set 2 is a 3D vector field. a) A slice of the volume with lines following the flow field. b) A volumetric representation of the surface extracted by the border detection algorithm.

way to test our vector field extraction filters and 3D filtering in real-time. Four filters were applied to the test set. Gradients were calculated by central difference and rotated 90 degrees clockwise around the z-axis. Flow vectors with a negative x-component were turned 180 degrees. Then a Gaussian blur filter was used before a median filter smoothed the flow field even more. At last we applied the filter that contains the border detection algorithm. The result is seen in Figure 11, and shows that some borders are detected also in places where the layers appear parallel.

The last test set is again generated from stacking copies of an image. We used an image that represents a seismic image picturing many sedimentary units (Colour plate Figure 12a). The data set is of size $512 \times 512 \times 64$. First the flow vectors were found by central difference as with test set number 3. Then the flow field was smoothed by Kang

et al.'s bilateral smoothing filter [8] reviewed in Section 2.2. Also, a 3×3 mean filter was used for more smoothing. All vectors were normalized. Colour plate Figure 12b is a rendering of the RGBA-vectors constituting the flow field. Colour plate Figure 12c and d shows the output of the border detection algorithm with two different threshold settings. The step size is 0.5 and the number of steps is 1000 in both cases. A path may evolve for less than 1000 steps if the path reaches the edge of the volume. The distance threshold in Colour plate Figure 12c is set so that the algorithm maps any seeding point where the seeded particles diverged for more than 16 units at the path ends. In Colour plate Figure 12d the threshold is set to 4 units. Clearly, a smaller threshold maps more points as a border point. The effect of changing the threshold value can be seen in real-time when the algorithm is run on one slice of the volume due to our GPU implementation. Therefore, the user can find a satisfactory threshold value before the entire volume is processed.

Testing was done on a machine with an *NVIDIA GeForce GXT 295* graphics card with a global memory of 896 MB and an *Intel^R CoreTM 2 Duo* processor with 2GB ram. Running times for the last and biggest test set of size $512 \times 512 \times 64$ was as follows: It took 2.4 seconds to filter the whole set by the three flow field extraction filters and running the seeding algorithm with 1000 path steps on one slice. When the seeding algorithm was run on all 64 slices, the same process took 44.5 seconds. The implementation was written in C++ and OpenGL within the framework of Volumeshop. All calculations are done on the GPU.

5 Conclusions and future work

The paper has demonstrated an automated method for highlighting unconformities in seismic data. An implementation of the technique, according to given implementation details, has shown a promising outcome in four different tests. Although the test sets were quite small, manually generated and more regularized than most seismic data sets, the presented implementation shows a possible way of detecting seismic unconformities on a global scale. We have also presented a programming pipeline for processing seismic data with all calculations done on the GPU. Two OpenGL 3D textures are used in a ping-pong fashion for reading and writing while filtering the data volume. The volume is updated in real-time when applying or adapting any filters.

For testing with actual seismic data sets, a few improvements to the program are needed. Seismic data sets are often very large in size, and it would not be possible to load such volumes in entirety onto the GPU. A method for processing large data sets in sequence, such as stream-processing, is necessary. We would also like to refine the detection algorithm itself to better handle the possible features of a flow extracted from seismic data.

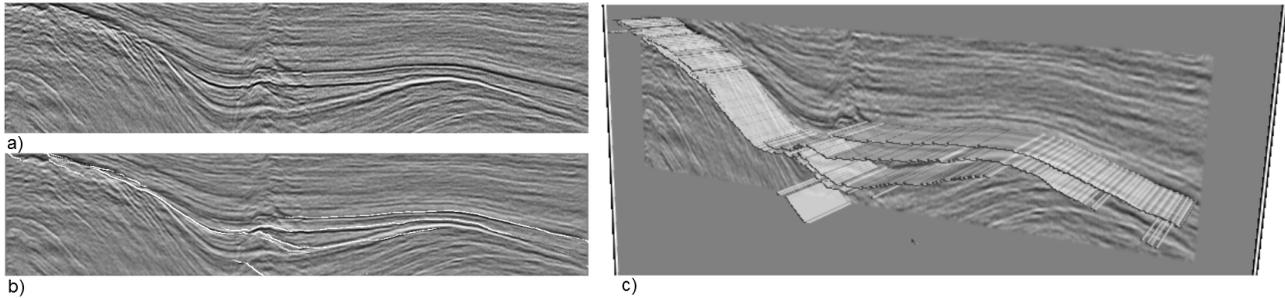


Figure 11: a) A seismic cross section. This image is copied and stacked to constitute a 3D test set. b) One slice of the output volume after our border detection algorithm is employed. The detected border is displayed as a white line. c) The detected border is seen as a surface when displayed in VolumeShop. A slice of the volume is added for reference.

Acknowledgement

Thanks to Danel Patel, who supervised this project, for all his help and inspiring discussions.

References

- [1] M. Bahorich and S. Farmer. The Coherence Cube. *The Leading Edge*, 14(October):1053–1058, 1995.
- [2] M. Brown and R. G. Clapp. Seismic pattern recognition via predictive signal/noise separation. *Stanford Exploration Project, Report 102*, pages 177–187, 1999.
- [3] S. Bruckner and M. E. Groller. VolumeShop: an interactive system for direct volume illustration. *IEEE Visualization '05*, pages 671–678, 2005.
- [4] O. Catuneanu. *Principles of sequence stratigraphy*. Elsevier, 2006.
- [5] S. Chopra. Coherence Cube and Beyond. *First Break*, 20(1):27–33, 2002.
- [6] S. Chopra and K. J. Marfurt. Seismic attributes - A historical perspective. *Geophysics*, 70(5):3–28, 2005.
- [7] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Prentice Hall, 2008.
- [8] H. Kang, S. Lee, and C. K. Chui. Flow-based image abstraction. *IEEE transactions on visualization and computer graphics*, 15(1):62–76, 2009.
- [9] M. Kass and A. P. Witkin. Analyzing oriented patterns. In *Proc. Ninth IJCAI*, pages 944–952, 1985.
- [10] H. Kermit. *Niels Stensen, 1638-1686: the scientist who was beatified*. Gracewing Publishing, 2003.
- [11] W. Y. Ma and B. S. Manjunath. EdgeFlow: a technique for boundary detection and image segmentation. *IEEE transactions on image processing*, 9(8):1375–88, January 2000.
- [12] K. J. Marfurt. Robust Estimates of 3D Reflector Dip and Azimuth. *Geophysics*, 71(4):P29–P40, 2006.
- [13] G. Nichols. *Sedimentology and stratigraphy*. John Wiley and Sons, 2009.
- [14] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The State of the Art in Flow Visualisation: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.
- [15] T. Randen, B. Reymond, H. I. Sjulstad, and L. Sonneland. New seismic attributes for automated stratigraphic facies boundary detection. In *SEG Expanded Abstracts, Soc. Exploration Geophys, INT 3.2*, volume 17, pages 628–631, 1998.
- [16] A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53:157–185, 1991.
- [17] M. T. Taner. Seismic attributes. *Canadian Society of Exploration Geophysicists Recorder*, 26(9):48–56, 2001.
- [18] T. (Shell Malaysia E&P) van Hoek, S. (Shell International E&P) Gesbert, and J. (Shell International E&P) Pickens. Geometric attributes for seismic stratigraphy interpretation. *SEG, The Leading Edge*, 29(9):1056–1065, 2010.