Flow-based Segmentation of Seismic Data

Kari Ringdal, University of Bergen

Abstract
Reflected seismic waves are used to find out what the earth looks like beneath the surface. This is particularly interesting to the petroleum industry. The first step in analyzing data from reflected seismic waves, is to find the borders between the underground layers of rock that originated at different times of the earth’s history. On each side of such a border the compositions of the sediments will be different, and the angularity of the layers commonly vary. This paper presents a technique that attempts to enhance the process of separating the sedimentary units in 2D seismic data. One of the difficulties of this task is that layers belonging to different units can appear parallel on a local scale. The presented automatic technique addresses the data globally. Methods from the fields of flow visualization and image segmentation is utilized. The paper describes methods for preprocessing the data, as well as implementation details for the border-detection algorithm. The technique has been successfully applied to two test data sets, and the test results are presented.

1. INTRODUCTION
Stratigraphy is the study of rock layers deposited in the earth. It has been a geological discipline ever since the 17th century, pioneered by 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 \cite{Kern03}. The particles would settle evenly in horizontal layers on the lake or ocean floor. Through all of earth’s history, layers of sedimentary rock have been formed as wind, water or ice has deposited matter that consolidated into rock at the bottom of a body of water. A break in the continuous deposit results in an unconformity, where successive layers of sediments from different times meet. Figure 1 illustrates different types of unconformities. They represent different geological events. An example is angular unconformity, which can occur after a time of lower water levels and erosion. Younger sediments deposited on a angular layer, will then produce angular strata. Another reason for this type of unconformity is orogenic activity, such as shifting of continental plates, that tilt a whole sequence of layers.

The geologists analyze the pattern around an unconformity to decode the missing time it represents. Two types of patterns occur above an unconformity - onlap and downlap, and three types of pattern occur below - truncation, toplap and apparent truncation. Sedimentary facies are bodies of sediment layers, distinguished by its appearance or composition, delimited by unconformities. For more information on unconformities and other stratigraphic concepts, the reader is referred to Gary Nichols book on sedimentology and stratigraphy \cite{Nic09}.

Figure 2: Sediments of different facies can be indistinguishable in local areas such as inside the circle.

A 3D dataset created by processing reflected seismic waves is typically large and noisy. The first step in analyzing the data is to separate the masses of strata of different age, and seismic attributes that will highlight geological or geophysical effects are welcoming aids \cite{IR05}. Seismic attributes that measure qualities such as continuity, frequency, amplitude and phase have been around for decades. Three-dimensional attributes, such as dip and azimuth, have also been in focus. However, the traditional methods have been limited by addressing the seismic data on a local scale. The fact that sediments belonging to different facies can be in-
Figure 1: a) Illustration of different unconformities - angular unconformity, disconformity, nonconformity and local unconformity. b) Stratified sediments that show three sedimentary facies separated by unconformities. c) Unconformities in real seismic data.

distinguishable in greater parts of the picture (see Figure 2), calls for global analyzing tools.

This paper introduces an automatic method that consider the data on a global scale when looking for unconformities in 2D seismic data. To demonstrate the feasibility of the technique, it is applied to two different synthetic data sets. The paper is structured as follows: related work is discussed in section 2. The processing pipeline will be described in section 3. Section 4 gives performance data, and presents application to the two test sets. Section 5 draws conclusions and discusses future developments.

2. RELATED WORK

There have been different attempts to automate the labor intensive manual techniques used for analyzing seismic data. This section will focus on previous approaches to the task of 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.

2.1. Seismic methods

Methods that trace the orientation field in seismic data and detect where the strata lines terminates are widespread [RS05]. Randen et al. [RRSS98] proposed a method where the dominating local orientation of each seismic sample is estimated by using vertical and horizontal partial derivative of Gaussian filters followed by directional smoothing. Starting at any sample in the extracted orientation field, a curve is generated in the direction of the local angle (see Figure 3). The curve form a flow line along the orientation field, and intersection points of flow lines are marked with a X. These points are likely to be on a bounding surface of an unconformity.

A recent paper by Hoek et al. [vHGP10] describes a new method for finding unconformities. Like previous methods,
an orientation field that is everywhere perpendicular to the seismic reflections is used. This normal vector field, sometimes referred to as the seismic dip/azimuth field, is estimated by first using Gaussian derivative filters. Then the structure tensor field is found and regularized. Finally the principal eigenvector of the structure tensor is extracted. Similar to the method by Randen et al., the unconformity attribute by Hoek et al. highlights bounding surfaces (e.g. onlap and downlap surfaces as well as angular unconformities). Here 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 density of flow of the dip field in a neighborhood of a predefined size and is therefore still a local method that does not capture events taking place outside the neighborhood.

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.

Because of the apparent similarities to seismic images, the field of fingerprint image processing is relevant to our method. A lot of research has been done in fingerprint image enhancement, particularly when it comes to vector field extraction. Like in seismic data evaluation, it is essential that the extracted orientation-image represents the intrinsic property of the image and defines invariant coordinates for ridges and valleys in a local neighborhood. The gradients of a fingerprint image are the basis of the vector field extraction algorithms. The gradient of any point in a picture is the vector pointing in the direction of the greatest change. In image processing different operators are used for finding gradients. The Sobel operator is an example of an operator used for this purpose.

Hong et al. [HMWJ98] find the orientation field in a fingerprint image by first using an appropriate operator for finding gradients, and then dividing the picture into blocks the size of the operator mask. The local orientation of each block is estimated as orthogonal to the dominant direction of the Fourier spectrum of the same block. To eliminate noise and corrupted ridge and valley structures, Hong et al. convert the image into a continuous vector field so that low-pass filtering is possible.

Non photorealistic rendering (NPR) concern simplifying visual cues of an image to communicate certain aspects more effectively. Kang et al. [KLC09] 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 successfully employed. The paper of Kang et al. concern two areas that are interesting to our technique, namely the extraction of a vector field from an image, and extracting lines from isolated points. Kang et al. uses 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, to redirect weak edges, 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. presents this vector field extraction method as a successful base for extracting the main image features.

A different vector field extraction method is proposed by Ma and Manjunath [MM00]. They propagate edge flow vectors by utilizing a predictive coding model to identify and integrate the direction of change in color, texture and phase discontinuity at each image location. Edge energy and associated probabilities are computed for every pixel and represented by an edge flow vector. The vector points in the direction of finding the closest boundary pixel.

Edge linking is another interesting area from image pro-
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 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 \cite{GW08}. There are also global processing methods, like the Hough transform, that deal with linking edges. For Kang et al.’s flow-based image abstraction method \cite{KLC09}, part of the goal is to end up with a 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 are enhanced, and the good effect of this method is shown on grayscale, color and binary images (See Figure 4).

![Figure 4: a): Input image b): filtered by DoG filter c): filtered by Kang et al.’s flow-based DoG filter](image)

2.3. Flow field topology and extraction methods

Flow visualization is a subfield of data visualization that develop methods to make flow patterns in transparent fluids visible. In the last decades, this field has been growing steadily as computer performance has increased. Flow features and techniques for topology extraction of steady vector field data will be the focus of this section.

A feature is a fact or situation that happens in a data set. One can see it as a structure or an object that is of interest to the researcher. Shock waves and vortices are examples of flow features as well as boundary layers, recirculation zones, and attachment and separation lines (see Figure 5).

![Figure 5: Attachment and separation line features.](image)

Because of the layer-pattern of the sediments, features that are most likely to occur in a vector field extracted from seismic data, are separation and attachment lines. 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. deal with different methods for separation and attachment line extraction \cite{PVH03}. 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 paper, 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 2D seismic data follow the processing pipeline shown in Figure 6. 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 the focus will mainly be at the second step of the pipeline - the mapping of stratigraphic surface probability. In this step lies the novelty of our method.

3.1. Preprocessing - vector field extraction

As described in the Related work section, there already exists efficient methods for vector field extraction. It is important to underline that the first step of the pipeline is an important step, because a regular vector field, that is a good representation of the data, is crucial to our technique. Although the idea is to use the vector field extraction methods described in the Related work section, these were not implemented for our tests. Since there are well tested methods for the task of vector field extraction that are widely used within the fields of seismic stratigraphy and image processing, we did not see the need to implement this at the current state of our research. The used test sets are manually generated vector fields.

After the vector field extraction, the vectors are normalized and, if necessary, turned 180 degrees so that all vectors have a direction within the same 180 degree sector. This vector field is from now on referred to as the flow field, or just the flow when a part of the flow field is in reference. The
flow has a direction parallel to the sediment layers it represents, and it is perpendicular to the seismic reflections. The flow field is normalized to make sure the flow "moves" at the same speed at every point. Additionally, the flow is also saved in its opposite direction in the preprocessing step. This flow field is referred to as the opposite flow field. Two one-dimensional arrays of vector objects store the flow field and the opposite flow field in memory.

3.2. Mapping of stratigraphic surface probability

Our technique use particle seeding as is common within the field of flow visualization. However, the paths of the particles are not visualized. The user defines the resolution of a grid of seeding points, and for every seeding point two particles are seeded - one in each flow direction. The particle seeding is stored in memory as a one-dimensional array. The structure of the array is a seed structure that holds the seed point and the end points of two paths that are generated. The seed structure also holds the number of steps that are performed in each path direction. The paths of the particles are found by the use of the Runge Kutta 4th order (RK4) method. The paths are followed until the maximum number of steps is reached or the path hits an edge. The flow vectors of every step of the path are found by bilinear interpolation in the flow field.

If the RK4 integration method uses step size $h$, 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. Both the step size and the number of steps can be adjusted by the user. The step size parameter together with the number of steps parameter 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.

After the seeding is done, the algorithm runs though the seeds and checks the distance of the end points to the end points of the nearest neighbor seeds. Four neighbors are checked - north, east, west and south, but not the diagonals.

© The Eurographics Association 2010.
The distances of the middle point to its nearest neighbors are found at the end of the paths of each seed. These distances are marked with a blue line in Figure 7. The greatest distance is saved in a one-dimensional array with the same index as the middle seed of the five seeding points (marked as Probable surface point in Figure 7). Basically, this step generates an array containing a distance value for every seed point. Edge points are disregarded and set to zero. The array is later used as a look-up table when mapping the probable surface points.

Five seed points have been chosen for the process of generating the distances because of the symmetry of this constellation and its ability to detect edges in any direction. To make sure the surface mapping will also happen close to the stellation and its ability to detect edges in any direction. To make sure the surface mapping will also happen close to the edges of the vector field, certain measures can be taken. Dividing the distance \( d \) between two particles by the sum of the number of steps \( s_1 \) and \( s_2 \) of the particles, gives a value in the interval \([0, 1]\) (as long as the step size is 1 or smaller). This value can again be multiplied with the distance \( d \), and it makes sure that a path of few steps will be as sensitive to a threshold as a path of many steps. The adjusted distance \( d_{adj} \) is given by:

\[
d_{adj} = d \cdot \frac{d}{s_1 + s_2}
\]  

Even without distance adjustments, the process of generating the distances can be considered a bottleneck in our algorithm. Although it is not as computational heavy as the RK4 process, it will be significant when run on a large dataset.

In the next step of our method, the unconformities mapping is done by comparing the distance values from the look-up table to a given threshold. If a distance value exceeds the threshold, its corresponding position is marked as a black point in an OpenGL texture. The found points will constitute a surface line when the algorithm is done. It is not a given that the outcome is a finished segmentation of the data, edge linking may be necessary.

The texture with the probable surface points/lines is blended with a visualization of the vector field. The vector field is represented by short lines drawn with the same angle as the flow but without arrow heads to indicate a direction. This is done intentionally since the flow is used in both directions, and also, the layers the vectors represent are directionless.

If the user decides to adjust the threshold, no new calculations are needed. Only the previously generated look-up table is used. The threshold can therefore be interactively adjusted to a value that best show the unconformities. This will be a valuable feature for finding good default settings when dealing with actual seismic data.

3.3. The segmentation process

The ultimate goal is that the task of segmenting seismic data into sedimentary units is fully automated. Should the presented technique be a complete segmentation process, edge linking is required. This is the last step of our pipeline. 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, two simple vector fields were generated. The first test set of size 100x100, simulates two sedimentary units that have parallel layers in greater parts of the picture (see Figure 8). As expected, the unconformity surface was only found in the area without parallel layers when the algorithm was run locally. Each particle path had 20 steps with a step size of 1. With a seeding-grid resolution of 256x256, it suggests that every particle traveled within a local neighborhood. The bottom picture of Figure 8 show 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.

Test set number two has a size of 256x256. It was generated from an image of stratified sediments. This is closer to a seismic data set, but still a simplification without any noise or other corruptions. As Figure 9 shows, our method found the unconformity surfaces in this data set. The figure is also an example where edge linking is necessary for a complete segmentation. Adjustments of the parameters and threshold value would in this case map the unconformity all the way without edge linking. However, when dealing with real seismic data, edge linking could improve the outcome.

For actual running times of the algorithm, it was tested on a machine with 2 GB memory and an Intel dual core 3GHz processor. The 256x256 test set was used with a seeding grid resolution of 256x256. The RK4 parameters were set to 200 steps with a step size of 1. It took 10.07 seconds to complete all calculations and 13.2 MiB of memory were used. When the grid resolution was lowered to 128x128, calculation times was 2.75 seconds. The program was written in C++ with all calculations done on the CPU.

Probably, it is not necessary to use the RK4 method for a successful outcome of the presented test sets. A simpler integration method would be sufficient and speed up the algorithm. The choice to use RK4 is done in terms of the more complex flow fields that actual seismic data would generate.

5. CONCLUSIONS

The paper has demonstrated an automated method for highlighting unconformities in seismic data. An implementation of the technique, according to given implementation details,
Kari Ringdal, University of Bergen / Flow-based Segmentation of Seismic Data

Figure 8: Test set 1. above) An example of running the algorithm locally - each particle is followed for only 20 steps. below) The algorithm is run globally - each particle is followed for 200 steps.

The presented implementation shows a possible way of analyzing seismic data on a global scale. Two bottlenecks became apparent in the implemented program. The first is the RK4 calculations, and the second is the process of finding distances between the seeded particles. Decreasing the calculation times in this last bottleneck would benefit the user interaction when changing the size of the seeding grid or RK4 parameters. A possibility is to use a triangle constellation instead of the five particles we have used. Then only four distance values (two at each path end) would have to be calculated and compared for every seed point instead of eight. However, at this point it is not clear if three particles instead of five would be sufficient when dealing with real seismic data.

For testing with actual seismic data sets, a few improvements to the program are needed. The dimensions of such data sets require better performance when it comes to speed.

Figure 9: Test set 2. above) The algorithm is run on a vector field that was manually extracted from a picture of stratified sediments. below) Results from the above test blended with the input picture.
and memory allocation. To solve memory issues raised by a big data set, the seed data-structure should be serializable and have save and load methods so that it can be stored in a file. Since the calculations in the algorithm are done for each seed point without depending on the outcome of each other, it could be calculated in parallels. This makes the program a good candidate for a GPU based program. If calculations were done on the GPU, big seismic data sets could most likely be processed with the presented algorithm.

ACKNOWLEDGEMENTS

Thanks to Daniel Patel for supervising this project.

References


© The Eurographics Association 2010.