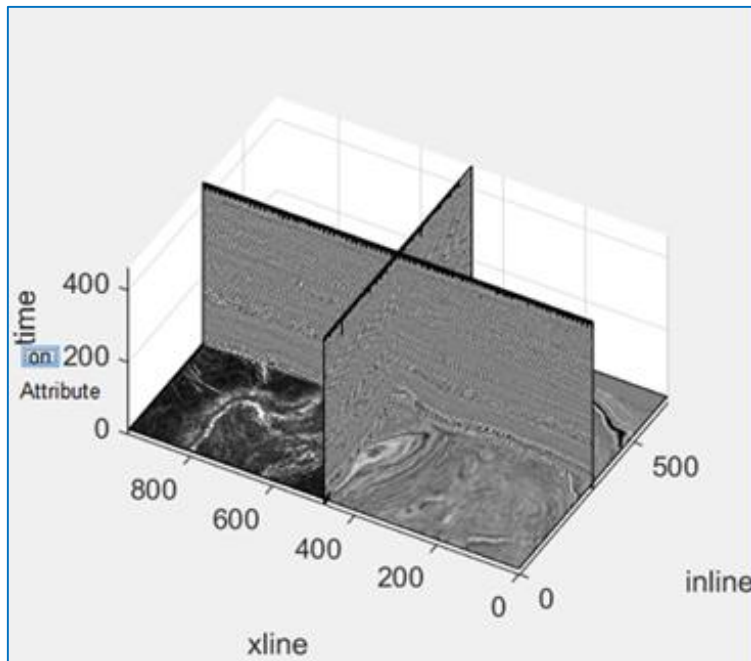




Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: Master in Computer Science	Spring semester, 2015 Open / Restricted access
Writer: David Sundli Barka (Writer's signature)
Faculty supervisor: Tom Ryen External supervisor(s): Ahmed Adnan Aqrawi	
Thesis title: Sobel Based Edge Detection Algorithm with Adaptive Operator Size Applied to Post-Stack Seismic Data	
Credits (ECTS):	
Key words: Seismic Attribute, Image Processing, Sobel filter, Adaptive filter, Frequency Analysis, post-stack seismic data, edge detection, dip guiding	Pages: ...69..... + enclosure: DVD & zip file Stavanger, ...12.06.2015..... Date/year



Sobel Based Edge Detection Algorithm with Adaptive Operator Size Applied to Post-Stack Seismic Data

UNIVERSITY OF STAVANGER
DAVID BARKA

ABSTRACT

Exploring the subsurface is today mostly done by researching seismic imaging thoroughly before attempting to bore exploration wells. By applying seismic attributes (imaging filters), we can highlight certain aspects of the seismic, which will help geoscientists to determine with better certainty potential reservoirs. Faults are an important feature in reservoir identification as oil is likely to emigrate via them. By highlighting faults with seismic attribute, it is possible to create three-dimensional models and through advance simulations a better understanding is achieved. However, achieving a detailed image of a faulted area can be difficult as larger faults can overshadow minor faults, parallel or perpendicular. This is especially hard considering a static filter operator size that can be too large to pick up the smaller faults.

In this work, we propose to adjust the filter operator size according to the frequency content of the seismic, by adapting the operator size according to the current filter position. We aim to achieve higher amounts of detail in fault detection using a method based on the Sobel method. We will also investigate the chaotic nature of the seismic to determine if the use of dip correction is needed. These are all novel methods in the approach to fault estimation in seismic.

Our implementation is done in MATLAB and tested on a dataset in the North Sea offshore Netherlands, F3, which has a heavily faulted area due to salt tectonics. We perform a vertical frequency analysis using the discrete Fourier transform to determine an operator size that is used in a horizontal frequency analysis using variance. To avoid picking up stratigraphy, we use dip guiding that let us filter along the different layers of the earth. This helps avoid geological noise detection. However, in chaotic areas, such as salt domes, we avoid using dip guided filtering, as there are no layers to follow, leading to a possible false representation. With the use of linear interpolation, we want to capture small details that otherwise would be absent. To better represent the faults in our image, we apply a vertical mean smoothing filter with neighboring values. This works due to the vertical nature of fault features. Evaluating all the stages in our algorithm, we find that the computational complexity is bounded by $O(N^2)$.

The approach of an adaptive operator size for a Sobel based edge detection to achieve a more detailed image proves to be successful as it surpasses the industry standard for fault detection. Some of the larger faults are weaker in value than in Petrel. However, the noise in between the faults has been reduced substantially and smaller faults parallel and perpendicular have emerged. Ultimately giving a more realistic and complete representation of the fault networks present in the sub-surface.

ACKNOWLEDGEMENT

This thesis is the result of a Master thesis assigned from Schlumberger Norway and its employee Ahmed Adnan Aqrabi, which has been of great help throughout the entire thesis. Geological, seismic attribute explanations and general guidance (during thesis as well as from the start at the university) has been greatly appreciated, I would not be in this position today if it were not for him.

I would like to thank my supervisor Tom Ryen for his dedication and enthusiasm to all of his students and his frequent check up to ensure success for all his students. I would also like to thank to my fellow students Thomas Hinna and Morten Salte that I have worked with during my years as a student.

A special thanks to my family for their support from day one in my pursuit of a higher degree and making my student life a comfortable one.

David Sundli Barka

Stavanger

June 2015

CONTENTS

Abstract	ii
Acknowledgement	iii
1 Introduction	1
1.1 Goals	2
1.2 Our Contribution	2
1.3 Thesis Outline	2
2 General Background	5
2.1 Seismic Imaging and Structure	5
2.1.1 Seismic Imaging	5
2.2 Previous Work	8
2.3 Petrel	9
2.3.1 Petrel Geophysics Component	9
2.3.2 Petrel Exploration Geophysics	10
2.3.3 Petrel Seismic Interpretation	10
2.4 MATLAB	10
2.4.1 CREWES	11
2.5 Dataset	11
3 Algorithmic Background	13
3.1 Fourier Transform	13
3.1.1 Fast Fourier Transform and Discrete Fourier Transform	14
3.2 Noise Removal	14
3.2.1 Image Filtering	15
3.2.2 Mean Filter	15
3.2.3 Gaussian Filter	16
3.2.4 Median Filter	17
3.3 Edge Detection	17
3.3.1 Sobel Filter	17
3.4 Dip Guiding	18

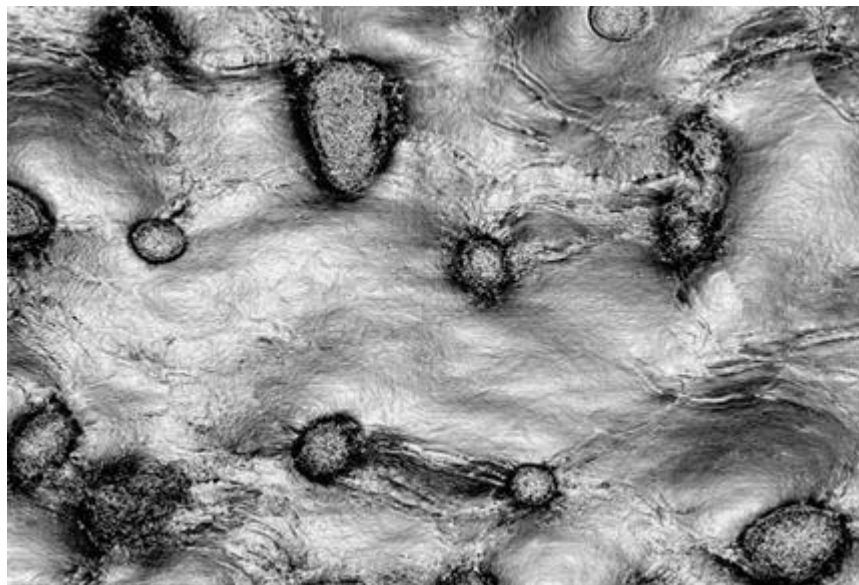
3.5 Adaptive Operator Size by Chaos Analysis	19
4 Methodology and Implementation	21
4.1 Overall View	21
4.1.1 Three-Dimensional Seismic Viewer	22
4.2 Frequency Analysis	23
4.3 Dip Straightening	25
4.4 Variance Analysis	27
4.5 Interpolation	28
4.6 Sobel Edge Enhancement	30
4.7 Vertical Smoothing	31
5 Results and Analysis	33
5.1 Sobel on Pure Seismic Data	33
5.2 Sobel with Dip Guiding	35
5.3 Sobel with Dip Guiding and Interpolation	36
5.4 Vertical Smoothing	39
5.5 Comparison with Petrel Results	41
5.6 Gaussian Smoothing Pre-Processing	44
5.7 MATLAB Algorithm Run Time	44
6 Conclusion and Future Work	47
6.1 Conclusion	47
6.2 Future Work	48
Appendix A	49
Appendix B	49
Appendix C	52
Bibliography	58

CHAPTER 1

1 INTRODUCTION

In today's oil and gas industry exploring subsurface areas is mostly done by researching seismic imaging thoroughly before attempting to bore. To achieve the most information from seismic imaging, different types of image filtering are applied; which are called seismic attributes (see Figure 1) in the industry. This will help geoscientists to determine with better certainty locate potential reservoirs. However, when applying seismic attributes of any kind, the operator size of the attribute is either predetermined or user selected in a static fashion. By doing this, important details can be lost in the process that can be decisive when drilling for oil. If the operator size is too large, details of the image will be lost, such as small faults or corrupt salt domes. However, if the operator size were too small, the attribute would render ineffective; the image would remain unchanged. Because the seismic structure is built from an array of fluctuating signals ranging from lower frequency in thinner rock layers, to higher frequency in thicker rock layers. It is natural to think that a static operator size would be less than ideal to keep thinner and thicker layers from merging. Over the years, when applying seismic attributes, users have decided parts of the seismic to focus on and made an educated guess on the operator size. Thus, resulting with a multitude of equal datasets that each has their selected sections filtered.

Figure 1 Innovative illumination attributes for a new look at complex structural delineation [1].



The frequency information in the image can be found by using the Fourier transform. We will use Fourier transform to translate the three dimensional space x , y and z into the occurrence of frequencies. We can then use the occurrence of frequencies to help determine the operator size of the seismic attribute.

1.1 GOALS

In this work, we want to adjust the operator size according to the frequency that is the most dominant around the filter operation site. The essence is to challenge the common approach in the oil and gas industry and do a comparison study with the leading edge detection attribute from the last decade. To achieve this, we have two main goals:

- To develop an algorithm that analyses around the seismic attribute center to determine the right operator size.
- Use directional filtering along derivative plane to minimize noise detection.
- To implement edge detection that uses the algorithm mentioned above.

1.2 OUR CONTRIBUTION

The contribution of this thesis are as follows, it will change the way seismic attributes are applied throughout the industry. It will show that adaptive operator size for seismic attributes are the way forward.

- Contributes are the mentioned in goals, and the results.
- A more detailed image of edge detection.
- Building upon our previous work in adaptive seismic attributes [2] [3] [4].

1.3 THESIS OUTLINE

This thesis will be structured as follows:

Chapter 2: Relevant background material is highlighted and explained for readers that have no prior knowledge in topics such as: seismic imaging, Petrel software and MATLAB.

Chapter 3: Relevant background of the algorithms that we will build our work on. This include frequency analysis, as in Fourier transform, Noise removal with smoothing filters such as mean and Gaussian. Edge detection using Sobel operators. Enhanced attribute performance by using dip guiding. We will also look at Adaptive operator size by chaos analysis.

Chapter 4: Description of the implementation in this thesis, how it was performed and why certain implementations decisions were made. One will also find the different stages that we went through during the development of this algorithm.

Chapter 5: Presentation and discussion of the results from different algorithms. We will mainly focus on the implementation and the different results when adjusting the parameters of the algorithm.

Chapter 6: In this chapter, we will present the conclusion of the thesis and some future work that can be done to further improve upon this thesis.

Appendix A: An outline of the MATLAB source code that was developed during this thesis.

Appendix B: Presentation poster that was held at the University of Stavanger on June 1.

Appendix C: Paper on Adaptive Sobel Based Edge Detection for enhanced Fault Segmentation published in IPTC 2014

CHAPTER 2

2 GENERAL BACKGROUND

In this chapter, we will introduce background material in topics that are relevant to our work. The reader can skip this section that have prior knowledge. The chapter will be divided into four sections.

In section 2.1, we will present Seismic Imaging and the connection to Image Processing. In addition to an explanation of the Seismic Structure and how the filtering process is applied, and what challenges this will introduce for further interpretation of the data.

In section 2.2, we will talk about previous work that have been done in this field. Such as frequency analysis, Sobel based filters and adaptive operator size.

In section 2.3, we will introduce the Petrel Software and its features as an integrated workflow tool that includes most of the reservoir modeling procedure. Some data used for this thesis (namely Dip guiding, see 3.4 Dip Guiding for more information) are preprocessed using Petrel.

In the last section, section 2.4, we will introduce MATLAB, Image Processing toolkit, CREWES toolkit and our motivation to implement our algorithm there.

2.1 SEISMIC IMAGING AND STRUCTURE

In this section, we will explain the definition of seismic imaging and the connection to Image Processing. Then we will focus on our motivation in this work by explaining the function of applying filters on the seismic data and what challenges this introduce. This is the main inspiration for our work.

2.1.1 SEISMIC IMAGING

Seismic images can show significant detailed picture of the earth's subsurface geology [5]. These pictures can be used to uncover structural features at large depths giving a better understanding of how to proceed to identify locations of oil and gas repositories. Seismic Imaging sends a forceful sound source into the ground where each segment of the earth's layer reflects some sound waves back "echoes", which are received by geophones, analogous to microphones. The echoes intensity and time are used to estimate the subsurface conditions on computers. See Figure 2 and Figure 3.

Figure 2 Propagating sound sources [6]

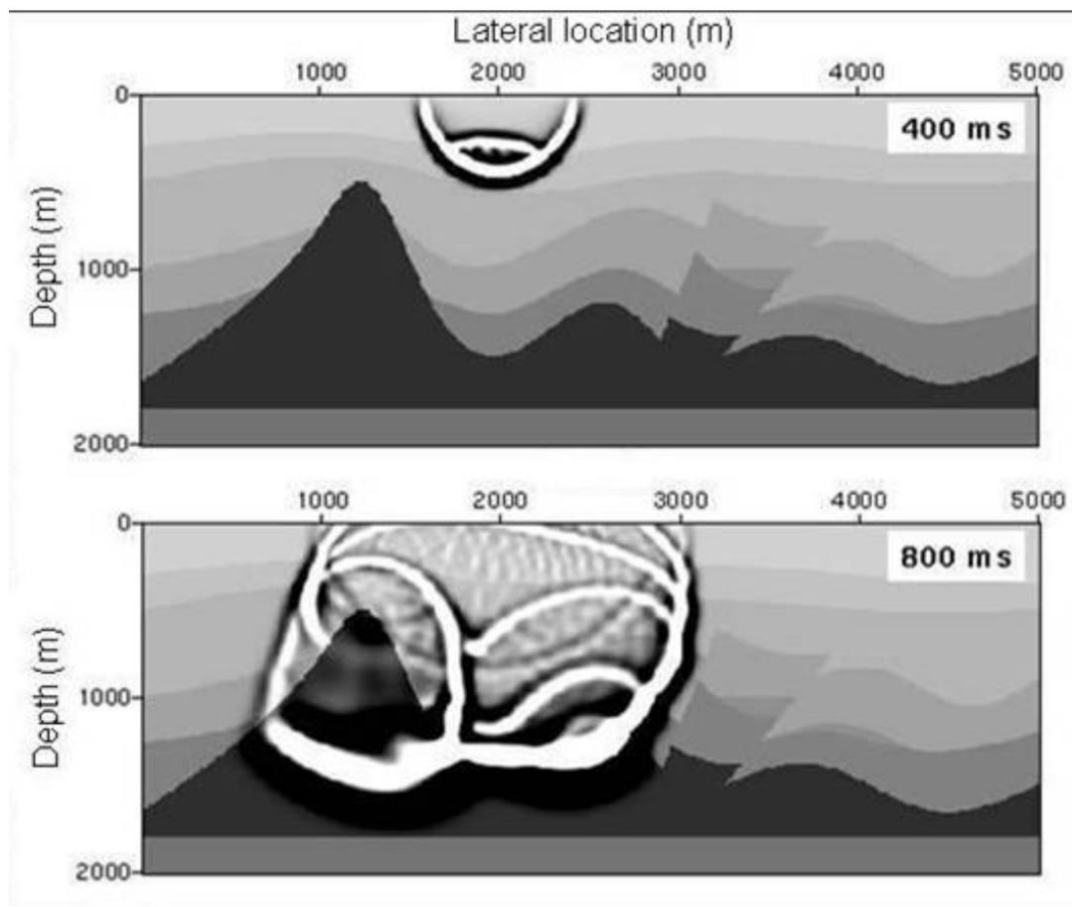
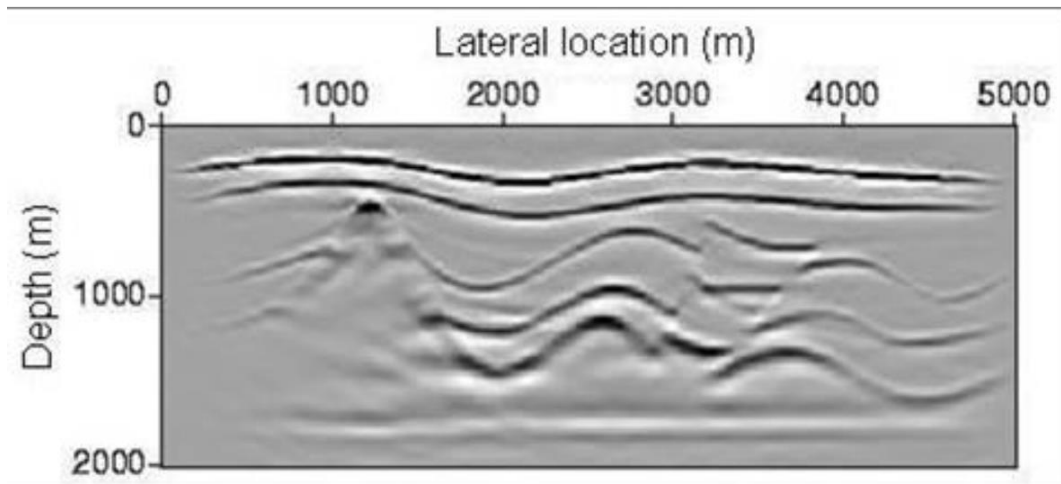


Figure 3 Processed data from acquisition from [6]

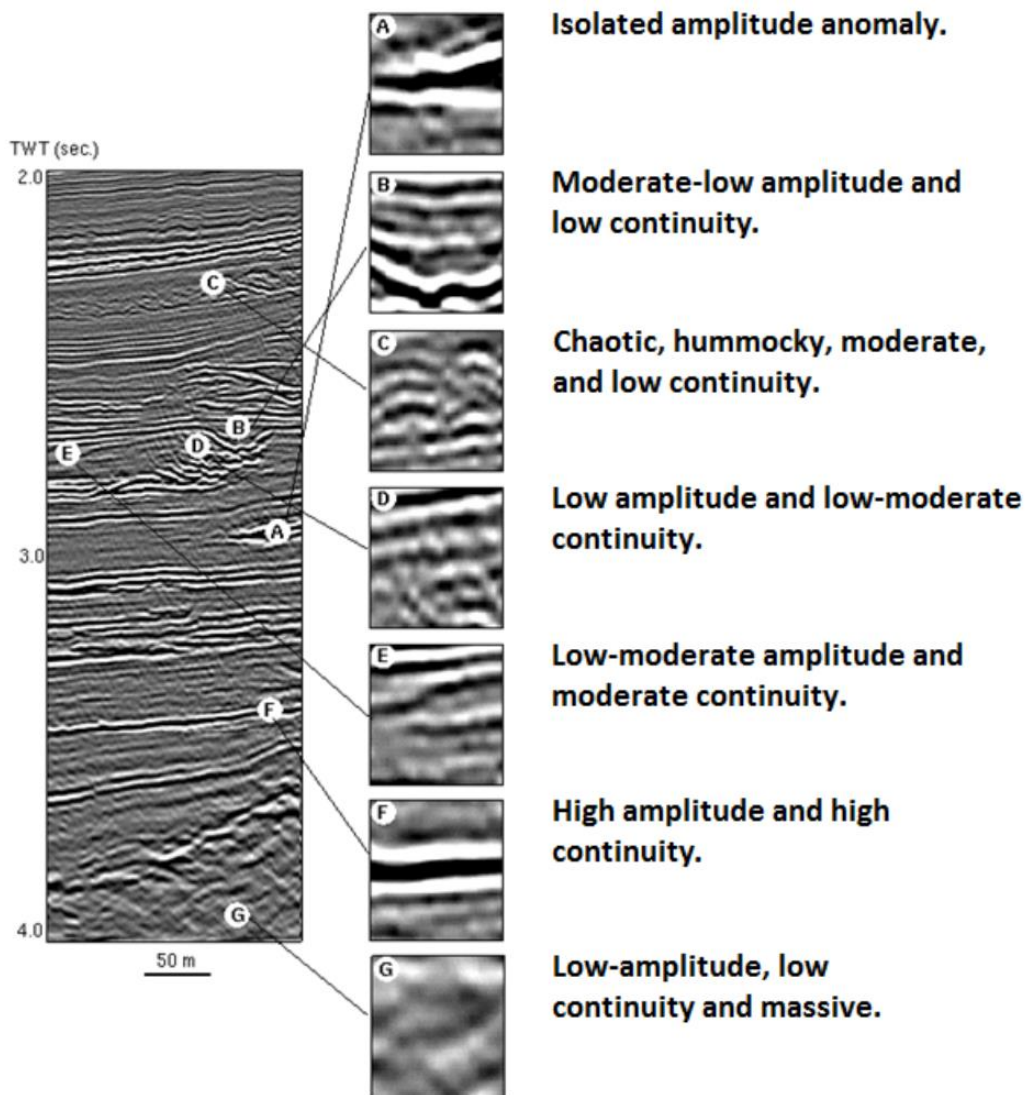


The data before data processing is called pre-stack. This is raw data that are memory space costly. To save space and computational power, an operation called stacking is used to average the recorded traces along the appropriate range of the four spatial axes. To further reduce computational power for analytical procedure, an approximate full pre-stack migration procedure is done, which is often called post-

stack imaging. Post-stack data provides a three-dimensional model of the subsurface structure which can be used to identify preferential flow paths. It determines the placement and screening of wells, and it help selecting a remediation technology.

Due to the depositional nature of the subsurface, the seismic images show patterns with a layered structure. In Figure 4, we can see seven different types of seismic textures [7] that can appear as a collection in a dataset. They may be easy to isolate in a two-dimensional seismic image. However, the move to the three-dimensional space makes it more difficult.

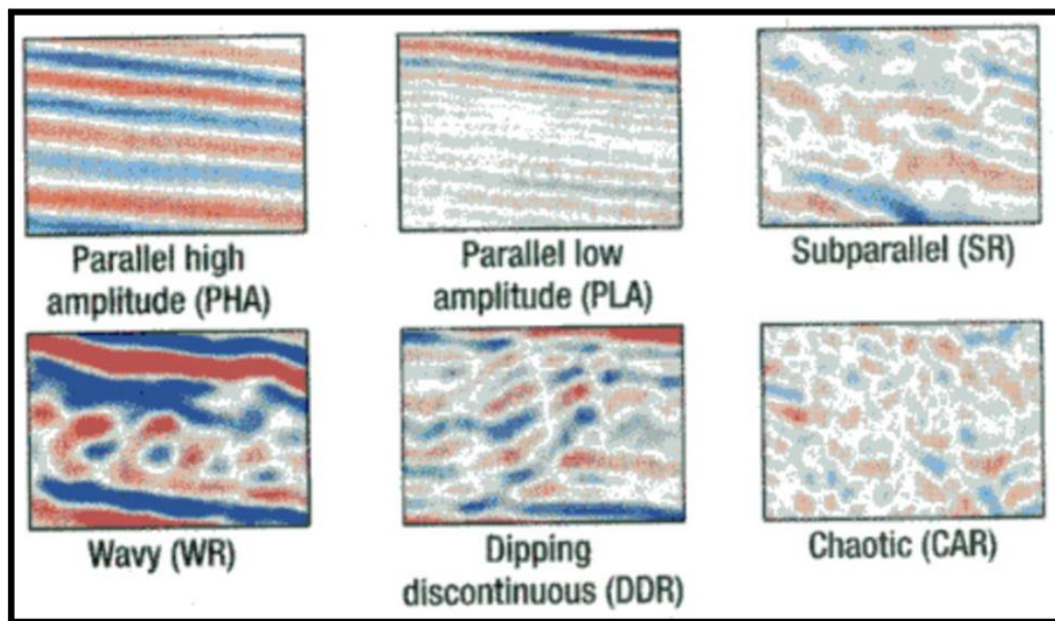
Figure 4 Seismic texture [7]



In Image Processing a texture is defined as a certain pattern regularity or structure. Textured layers can be split up in two parts in seismic imaging [8]. One part is the information of the signal perpendicular to the layered structure, which has a

characteristic frequency. It is determined by the change in the acoustic impedance of the subsurface rock combined with the seismic wavelet. This combined signal is described by using a time-frequency representation [9] [10]. The other part is geometrical information of the structure. In Figure 5, there is more focus on the chaotic nature of seismic data that can occur. If we point our attention on the four last samples, subparallel, wavy, dipping discontinuous and chaotic, it is important not to use a large operator size, as this will easily blend the different segments (blue and red) and create a misrepresentation of the data.

Figure 5 Seismic texture [8]



2.2 PREVIOUS WORK

Fourier transform, in this case discrete Fourier transform, is used to calculate a signal's frequency spectrum, and out of this the dominant frequency in signals. Another example is fast Fourier transformation convolution [9], an algorithm for convolving signals much faster than conventional methods.

Sobel filter (Sobel operator) [10] is a well known edge detection method. In 2011, a three-dimensional Sobel filter became an excellent way to determine faults in a seismic attribute [11].

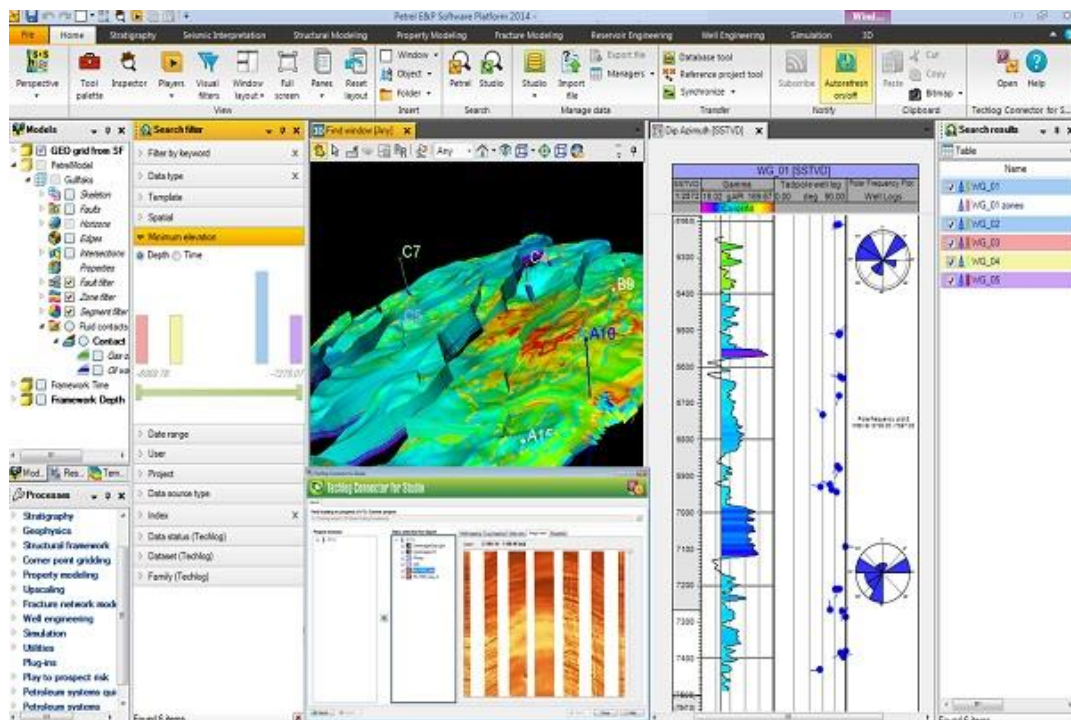
We have used adaptive operator before in seismic attributes with the use of chaos analysis and linear increase according to the depth. We have included a paper from the International Petroleum Technology Conference [2] in Appendix C. Other papers we have done are being withheld by Schlumberger at this time, and cannot be included in the Appendix [3] [4], which uses an adaptive operator size as a part of the algorithm.

2.3 PETREL

Petrel is an exploration and production platform (see Figure 6) that includes most of the reservoir modeling procedures and allows seamless collaborations between geoscientists. Petrel is the industry standard when it comes to edge detection. This is important as we are going to compare our results with Petrel's.

In this section, we will present the Petrel Platform with focus on seismic interpretation, geophysics component and exploration geophysics. The goal is to present the Petrel platform for readers that are not familiar with the subject. This software is the leading platform of its field and will be used for comparison to the results of this thesis. All information related to this section is provided by Schlumberger Information Solutions [12], [13] and [14].

Figure 6 Overview of Petrel 2014 [16]



2.3.1 PETREL GEOPHYSICS COMPONENT

Petrel provides a full spectrum of geophysical workflows, including two dimensional and three dimensional interpretation, a full set of surface and volume attributes which are used to identify faults and fractures, volume interpretation, convert domains and modeling while interpreting to enforce interpreters to build a structural framework. Petrel Geophysics support seismic data from different coordinate systems, from regional exploration to reservoir development.

2.3.2 PETREL EXPLORATION GEOPHYSICS

Petrel provides some key features for exploration geophysics; it can visualize and interpret regional two and three-dimensional seismic data manually or with advance auto-tracking techniques. Interactively create attribute maps of horizon or intervals. Petrel can interactively blend seismic volumes, isolates areas of interests, and instantaneously extract that is visualized into a three-dimensional object known as a Geobody. By generating and analyzing seismic attributes, Petrel can enhance the information that might be too subtle in traditional seismic, leading to a better interpretation of the data. The Z-axis of the seismic data can be converted back and forward between time and depth domain, bridging the gap between the time and depth domain. Petrel can perform an automated structural interpretation by focusing on the structural geology rather than conventional segment picking. This reduces the standard interpretation time while increasing level of geological detail and structural awareness giving a better understanding of reservoirs. By using neural network technology, it can estimate well logs, surfaces, seismic volumes and three dimensional property models.

2.3.3 PETREL SEISMIC INTERPRETATION

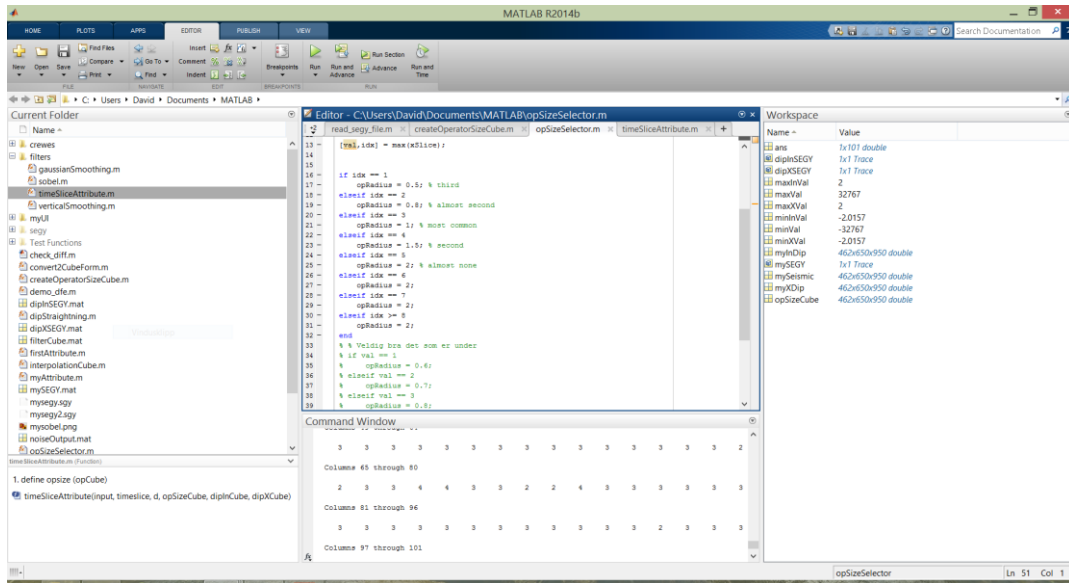
Petrel seamlessly combine the workflow of two-dimensional interpretation with the visual and performance benefits of three-dimensional volume interpretation. With these tools, it is simple to view and analyze the seismic volume in three-dimensional space. By applying structural attributes, different geological features can be highlighted for more certain results. A common task during seismic interpretation is extraction of faults and horizons into a three dimensional body, which also can be seen in Figure 6. Before this can be applied to a survey, it is crucial to pre-work the dataset to avoid a wrong segment to be connected or loss of faults.

2.4 MATLAB

In this section, we will introduce MATLAB (see Figure 7) and some added function sets and toolboxes that will be used during this thesis.

2 General Background

Figure 7 MATLAB 2014b



MATLAB [15], short for Matrix Laboratory, is a high-level and interactive environment used by millions of engineers and scientists worldwide. It is developed by MathWorks Inc. and containing an extensive documentation, tutorials and support. Another aspect with MATLAB is its analytical power for developing an algorithm that measures times of every function calls so that one can fine-tune every step of the algorithm.

2.4.1 CREWES

CREWES [16] (Consortium for Research in Elastic Wave Exploration Seismology) is an applied geophysical research group concentrating on the acquisition, analysis and interpretation of multicomponent seismic data. They have developed a set of MATLAB functions for seismic data operations. This is used to open and extract data from the industry standard SEG-Y files that is developed by the Society of Exploration Geophysicists (SEG) [17].

2.5 DATASET

For this thesis, we will use the F3 block dataset received from Opendtect.org [18]. This is an offshore dataset from the Dutch sector of the North Sea outside of the Netherlands. The block is covered in three-dimensional seismic that was acquired in the late 1980s. The dataset contains many diverse geological features such as Channels, Glacial plow marks, pockmarks, flat spots, Zechstein salt dome, surface expression and faults. These features will be of great use when testing our attributes and its effects.

The dataset has 650 elements wide (inline), 950 elements long (crossline) and a depth of 1848 milliseconds covering 384 square kilometer. The elemental values goes from -32767 to 32767.

CHAPTER 3

3 ALGORITHMIC BACKGROUND

In this chapter, we will introduce different algorithms that we will build upon in our work. We will go into the mathematical background on original algorithms that we will modify to use in our work with an adjacent explanation. This chapter contains five main sections.

In section 3.1, we will go through Fourier transform and its uses relevant to our work. Our focus will be on discrete Fourier transform.

In section 3.2, we will explain what noise reduction is in terms of seismic data as seismic data is made of noise.

In section 3.3, we will go through the operation of edge enhancement. In this case using the Sobel operator. Edge enhancement are used to highlight faults.

In section 3.4, the use of dip guiding and its function will be explained and how this will improve the output of the seismic attribute.

In the last section, section 3.5, we will show previous contributions related to our work. The first one is about adjusting the operator size according to chaos around the operator site and its depth [3] [4]. In this work, the goal of adapting the operator size fitting to the surroundings is the same as ours. However, we attend to use frequency in the seismic instead of looking at it as an image.

3.1 FOURIER TRANSFORM

The Fourier transform takes a function of time signal and decomposes it into frequencies, which it is built upon. All waveforms are the sum of sinusoids of different frequencies. The basic function of the Fourier transform can be seen in Equation 1.

Equation 1 The Fourier transform (a) and the inverse Fourier transform (b)

$$\mathcal{F}\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft} dt \quad (a)$$

$$\mathcal{F}^{-1}\{G(f)\} = g(t) = \int_{-\infty}^{\infty} G(f)e^{i2\pi ft} df \quad (b)$$

There are different approaches of implementing the Fourier transform such as Discrete Fourier Transform (DFT) and Discrete-time Fourier Transform (DTFT) the former is the one that we are focusing on.

3.1.1 FAST FOURIER TRANSFORM AND DISCRETE FOURIER TRANSFORM

Fast Fourier Transform (FFT) is an algorithm that computes the Discrete Fourier Transform (DFT) and its inverse. In this thesis, we have used the FFT function incorporated in MATLAB [19] (see Equation 2).

Equation 2 fast Fourier transform (a) and inverse fast Fourier transform (b)

$$X(k) = \sum_{n=1}^N x(n)\omega_N^{(n-1)(k-1)}, k = 1, \dots, N \quad (a)$$

$$x(n) = \left(\frac{1}{N}\right) \sum_{k=1}^N X(k)\omega_N^{-(n-1)(k-1)}, n = 1, \dots, N \quad (b)$$

Where

$$\omega_N = e^{(-2\pi i)/N}$$

Discrete Fourier transform is one of the most important discrete transform to perform Fourier analysis in applications [20]. The Discrete Fourier transform takes a finite list of equally spaced samples and converts them to a list of coefficients of a finite combination of complex sinusoids ordered by their frequencies with same sample values. In short, it can convert a sampled function of time or space to the frequency domain. By doing this, we can find the dominant frequency in a sample set with dimension of order 2^n , $n \in N$.

Because the input to DFT is expected to be of a periodic nature [21], likewise the output is expected to be periodic. However, this might not always be the case when the signal is either noisy or the signal length diverges from one period length. To bypass this potential problem, we can apply a window function: in our case the Hamming window. Hamming window is a bell shaped curve with the same number of samples as the DFT, where the first and last sample is close to zero.

3.2 NOISE REMOVAL

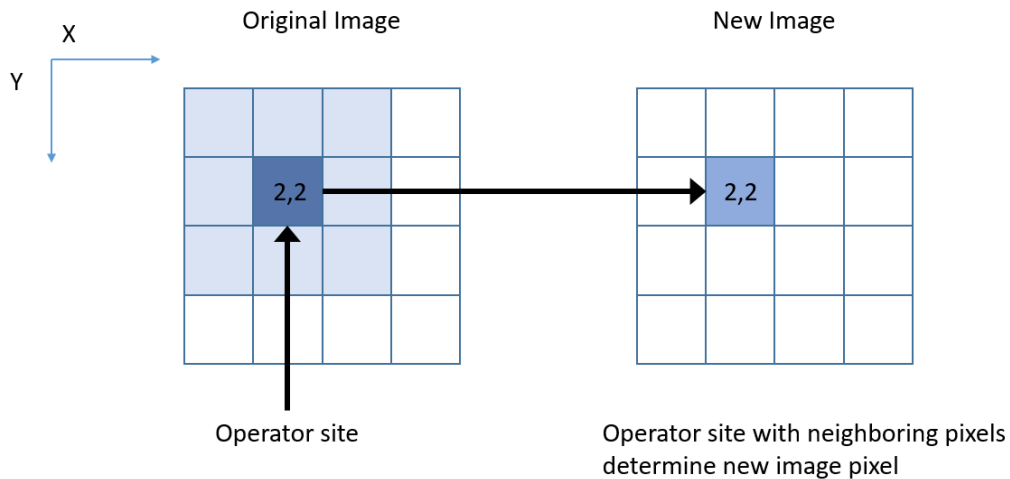
Because seismic data are retrieved signals from down to hundreds of meters below the earth's surface, it contains unwanted noise. As a general definition, any recorded energy that interferes with the desired signal can be considered as noise [22]. A result of noise can cause misinterpretation of edges that can be of high importance.

Unwanted signals (noise) can be produced in many different ways, likewise, there are many ways to remove or weaken its presence in an image.

3.2.1 IMAGE FILTERING

Before going through different image filters, one must understand how an image is filtered. When applying an image filter, it will traverse each pixel in a digital image that we will call the operator site. The operator site will look at the neighboring pixels (in Figure 8 we can see the light blue around the operator site which are the neighboring pixels called the window), and analyze the values in this window according to an algorithm to get a pixel value. This pixel value is then stored in a new image in the same x and y position, which in the end is the filtered image. This is to avoid values from formerly filtered pixels to influence the filter process of the remaining pixels.

Figure 8 Image filtering process



3.2.2 MEAN FILTER

Also known as box blur, a simple linear filter that uses a convolution matrix (kernel). A kernel represents the window of the filter, where one element overlap a single pixel in an image. In Equation 3 we can see a mean convolution matrix with size 3x3, the blue area on the left side showed in Figure 8.

Equation 3 Mean convolution matrix

$$Mean\ Kernel = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Each value in the convolution matrix represents a weight, how much each pixel in the window influences the final value represented in the new image. Usually, the sum of a convolution matrix equals one, in order to avoid over- or under saturation

of pixel values. In Figure 8 is an example of a mean value output, the eight light blue surrounding the darker blue are mixed making the output value a darker shade of blue.

3.2.3 GAUSSIAN FILTER

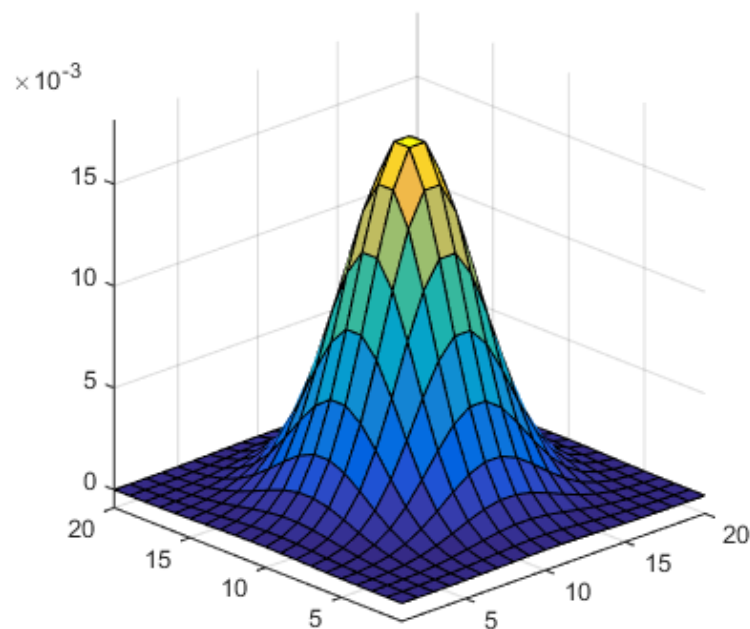
Named after the Gaussian curve, the two dimensional convolution matrix (see Equation 4) is weighted to form a three dimensional Gaussian bell (see Figure 9). Gaussian filtering has been studied in computer vision and image processing [23].

Equation 4 Two dimensional Gaussian convolution matrix

$$\text{Gaussian Kernel} = \begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6193 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix}$$

With Gaussian filter, the center pixel (operator site) has the most weight, which means it has the most influence on the end result preserving the structure of the image while smoothing. Even though it is used before edge detection to avoid finding unwanted edges because of noise [24], the use of Gaussian filter as preprocessing for edge detection may give displacement of edge position. In some cases edges vanish or phantom edges appear [25].

Figure 9 Three-dimensional Gaussian bell plotted in MATLAB



3.2.4 MEDIAN FILTER

A median filter is a non-linear filter that analyze the values contained in the image window, sorts them in numerical order from smallest to highest value, then returns the median value [26]. If the number of pixels results in an even number, the average of the two middle values would be returned. The median filter is optimal for removing impulsive noise (salt-and-pepper noise). On the other hand, fine details are also removed, as it does not see the difference between noise and fine details.

3.3 EDGE DETECTION

For structural analysis, using edge detection to highlight features of interest such as faults and salt dome borders.

3.3.1 SOBEL FILTER

Sobel filter, named after Irwin Sobel, who presented the idea of an isotropic 3x3 image gradient operator in 1968, is one of the standard image processing edge detection techniques [27] to measure luminosity of the onset of the red green blue (RGB) color space. It computes the rate of change across an edge as it is a first derivative operator.

Equation 5 Sobel convolution matrices in x and y axis (X and Y gradients)

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \text{ and } G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A$$

In Equation 5 we can see two convolution matrices that convolves (denoted by *) over an image (denoted by A) which equals images in an x-axis and y-axis image that contains the horizontal and vertical derivative approximations. G_x and G_y denotes the gradient directions of the Sobel filter, by applying Equation 6, we get the value that defines the edges in an image.

Equation 6 Gradients magnitude

$$G = \sqrt{G_x^2 + G_y^2}$$

In Figure 10 (a), we can see the image; cameraman, which is included in the MATLAB library. In (b) we have applied the Sobel filter and calculated the gradient magnitude for each pixel.

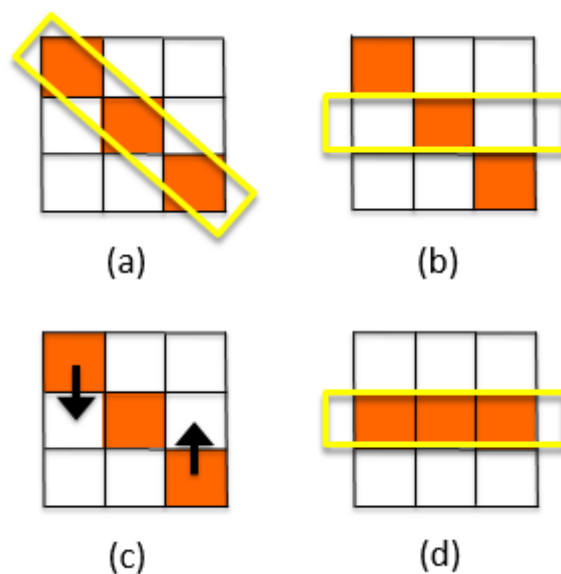
Figure 10 Image with and without Sobel filter



3.4 DIP GUIDING

In Schlumberger's Petrel software, there is a seismic attribute that calculates the curvature of the earth's layers [28]. In Figure 11, we can see a yellow box along the red squares (representation of pixels in an image) diagonally that is what we want to filter. If we apply an edge detection along the horizontal plane in (b), we would detect an edge along the red pixels. A way to solve this is to move the red pixels that are above and below the center, to make a straight line seen in (c) then apply a filter along the now straightened segment (d) and avoid unnecessary edge detections.

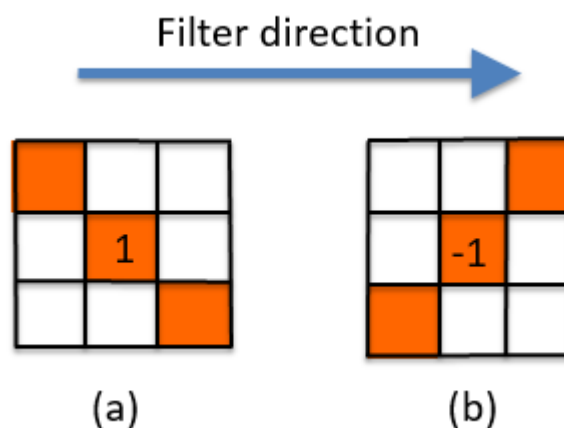
Figure 11 Example of filtering across segments versus filtering along segments



By applying the Dip guiding attribute in the inline and cross line, individual cubes are created in these directions showing the curvature with numbers usually between

1 and -1 (see Figure 12 (a) and (b) respectively). The value of one represents an angle of 45° either downwards (positive) or upwards (negative) in the filter direction which equal one pixel below or above respectively, hence, going against filter direction, results in an opposite angle (see Figure 12). Then the dip-guided cube can be used together with the original seismic data cube to move pixel values in an isolated state to produce similar values on a straight line before applying an image filter as illustrated in Figure 11 (d).

Figure 12 Segment angle according to pixel value



3.5 ADAPTIVE OPERATOR SIZE BY CHAOS ANALYSIS

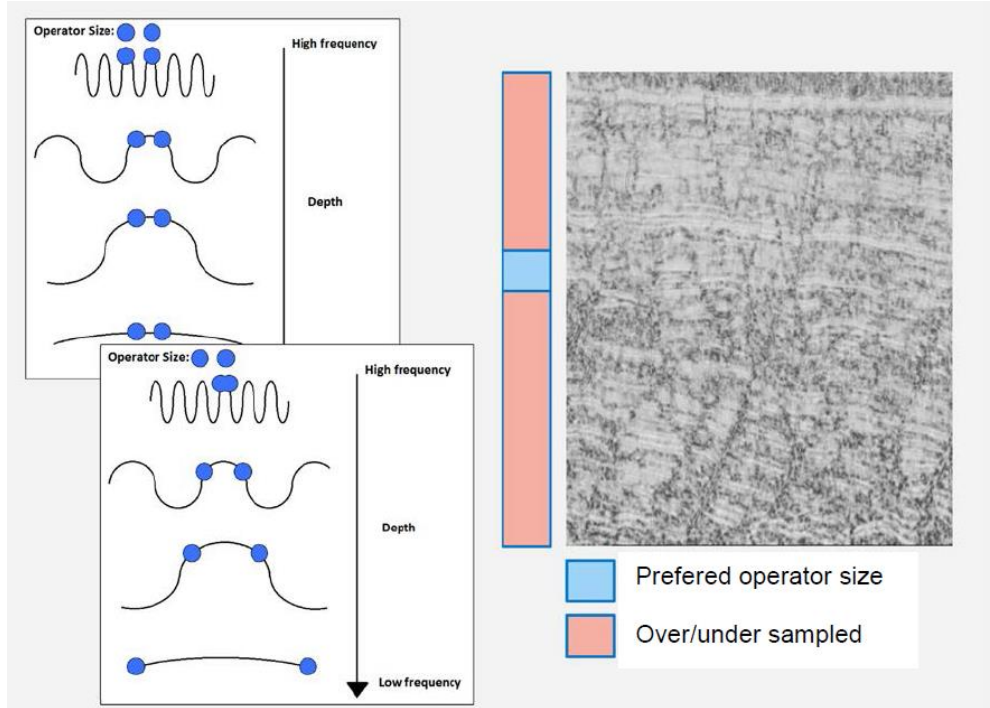
In previous work that we have done, using adaptive operator size for seismic attributes, depth and chaos analysis is used to determine operator size [3]. In this work, a three-dimensional Sobel based edge detector is implemented (Amplitude Contrast [11]). By looking at the nature of seismic data, it commonly changes from short wavelength to longer wavelength signals according to the depth/time. We then incrementally adjust the operator size adjacent to the depth/time, (see Figure 13).

However, seismic data is not that simple, Geological features such as dipping, salt and gas results in a chaotic and varying frequency regardless of which depth/time it occurs. Therefore, a textural analysis were introduced to take account for these changes in frequencies regardless of when it may occur, adapting the operator size on the fly. Changes to higher frequencies usually results in chaotic textures, therefore, chaos is used as seismic texture alteration indicator.

Results from this work indicated a higher level of detail of edge detection, highlighting smaller amplitude discontinuities better than a static operator size approach. It also proves to increase fault continuity and reduce the detection of noise.

Another related work using an adaptive attribute, is for noise reduction with hybrid filtering [4]. This work also uses chaos as a measure to adapt the operator size as well as filter type.

Figure 13 adjusting the operator size according to depth [2]



CHAPTER 4

4 METHODOLOGY AND IMPLEMENTATION

In this chapter, we will discuss the methodology in more detail than the previous chapter, and highlight certain implementation decisions. This chapter will be divided into seven sections.

In section 4.1, we will have a short description of our method. The sections that will follow will include more information about our method. We will talk about the user interface and how this will help simplify the analytical process. However, that is something we will talk more about in the next chapter.

In section 4.2, we will talk about discrete Fourier transform that we apply on the depth/time axis to choose an operator size for our next step.

In section 4.3, we will talk about variance analysis and how we further determine the operator size to be applied over the interpolation procedure.

In section 4.4, we will talk about how we use dip guiding in our model and the reason behind this.

In section 4.5, we will talk about interpolation between values according to the operator size.

In section 4.6, we will talk about the Sobel edge enhancement and why this is a good way to enrich visibility of faults.

In the last section, section 4.7, we will talk about vertical smoothing of the data as a final enhancement.

In the next chapter, we will discuss all the results related to these different methods.

4.1 OVERALL VIEW

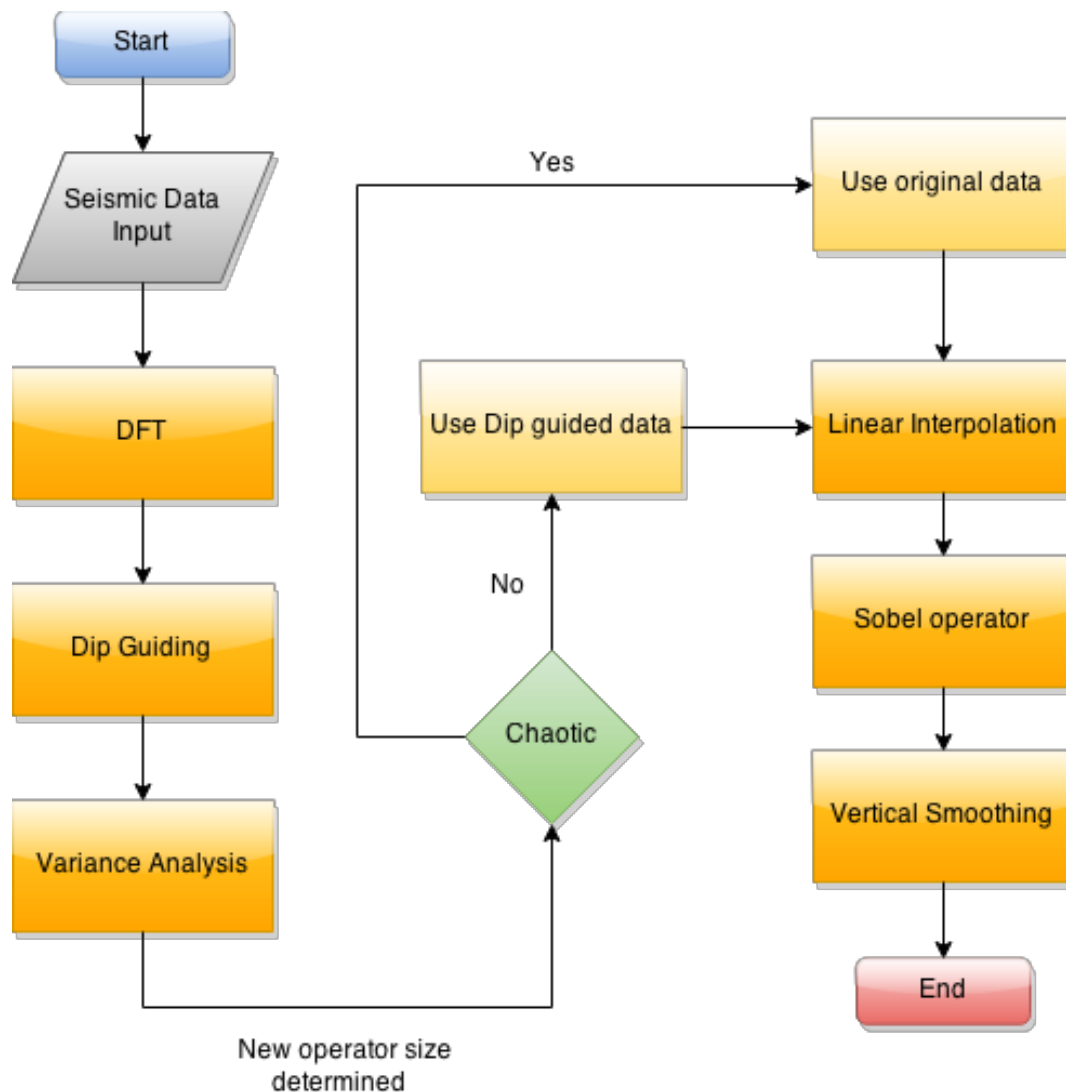
In this section, we will have a short description of the process that our algorithms will cover (see Figure 14 for an overview). We will divide the process into six stages:

1. Using Discrete Fourier transform in the depth direction for deciding operator size for each individual pixel in the dataset. For this process, eight samples are used with a Hamming window as overlay to achieve a periodic wave.
2. Dip guiding (straightening) sub cube data.
3. The ceiling value of each operator size value determines the radius of a variance analysis to determine a new operator size for the next step. The

variance result determines which data to use in next step, either the dip guided seismic data or the original seismic data.

4. The new operator size used for interpolating values into a three by three matrix.
5. The Sobel filter applied on the interpolated matrix, that is the value added to the output seismic cube.
6. After the Sobel operation is complete, a vertical smoothing applied to strengthen fault visibility and reduce noise elsewhere.

Figure 14 Algorithm flow chart

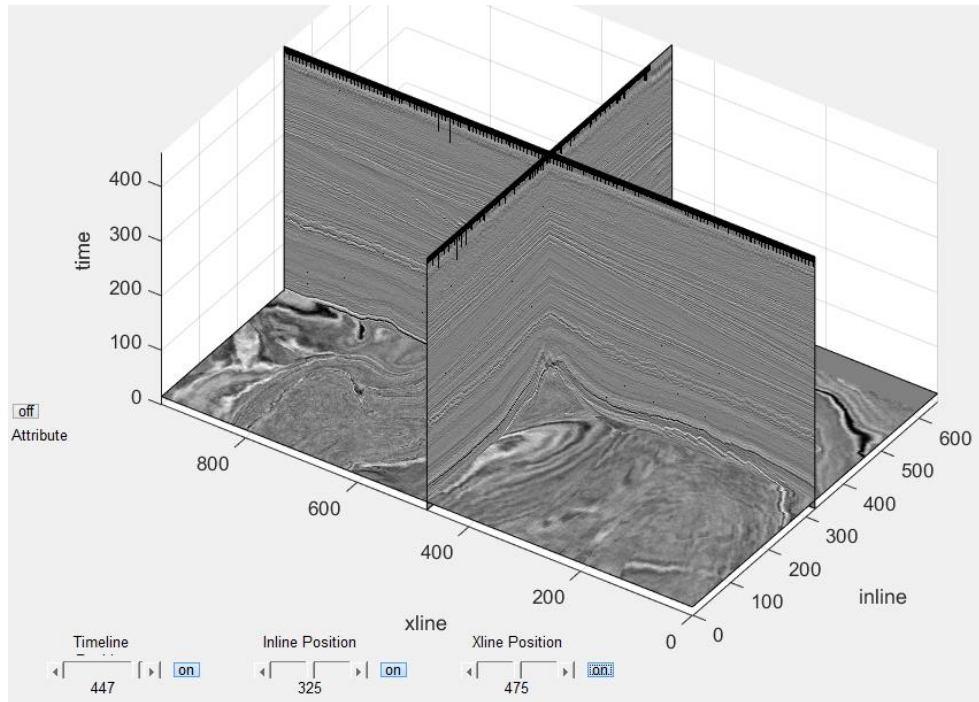


4.1.1 THREE-DIMENSIONAL SEISMIC VIEWER

To make it easier to view seismic data, we developed a three-dimensional seismic image viewer in MATLAB see Figure 15. It can show data slices in inline, cross line and depth/time. Three sliders on the bottom with corresponding buttons used to

activate intersection view and position in the seismic data, while the button on the left-hand side turns on the filter attribute on the time/depth-slice portion of the data. To get a better sense of position when viewing a single slice, the axis remains in all directions and not flattening as default.

Figure 15 The graphical user interface of the three-dimensional seismic image viewer.



4.2 FREQUENCY ANALYSIS

As a starting point for this algorithm, we perform a frequency analysis for each data point (pixel) in the dataset. By sending the seismic dataset to the `createOperatorSizeCube.m` (see Code snippet 1), a new cube with the same size as the input is generated. This cube contains filter operator sizes to be used in the variance testing process later in the algorithm. Creating an operator size cube can be done in advance or it is created during the algorithm.

Code snippet 1: createOperatorSizeCube.m

```

1 function opSizeCube = createOperatorSizeCube(input)
2 opSizeCube = input;
3
4 for k = 439:458
5     for i = 1:650
6         for j = 1:950
7             opSizeCube(k,i,j) = opSizeSelector(input(k-4:k+3,i,j));
8         end
9     end
10 end

```

If we look at Figure 16, the yellow square is the current position of the frequency analysis process. We take three samples above and four samples below including the current position giving us eight samples fulfilling the requirements of fast Fourier transform dimension of 2^n order (2, 4, 8, 16 etc.). These eight samples are sent to the opSizeSelector.m function (see Code snippet 2) for the operator size selection.

Code snippet 2: opSizeSelector.m

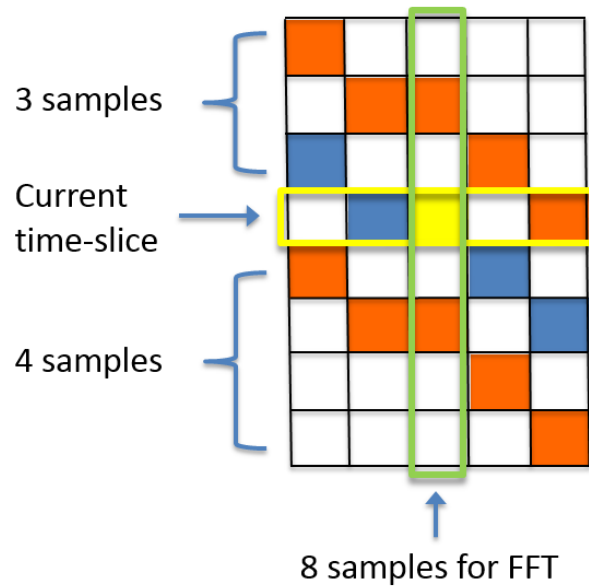
```

1 function opRadius = opSizeSelector(samples)
2 opRadius = 1;
3 d=length(samples);
4 fftGraph = abs(fft(samples)).*hamming(d);%hann(d);
5
6 [val,idx] = max(fftGraph);
7
8 if idx == 1
9     ...

```

In the opSizeSelector function, the array of samples are used in a fast Fourier transform with a Hamming window overlay. The operator size is determined by finding the index of the max value. Each index represents a value by selection of operator size.

Figure 16 Frequency analysis illustration



4.3 DIP STRAIGHTENING

In the `dipStraightning.m` function (see Code snippet 3), we send a sub-cube of the original seismic data, in-line- cross-line-cube. These sub-cubes are either three or five pixels wide each in the three axis, where the middle plane along the time-slice is the attribute depth and is displayed as the result (see Figure 17 (d) along the line of the red, blue and green squares). Looking at the attribute plane (viewed from above) (a), the blue square is the centroid of the cube. The value of one shows the angle of the neighboring values get straight segment line is 45 degrees (see (c), red square is 45 degrees above and the green square is 45 degrees below the blue). However, the neighboring pixels in the y-axis does not equal one, but 0.8 and 0.5. This would indicate that the neighboring pixels should be moved 36 and 22.5 degrees respectively to form a straight line. In our algorithm, we use the centroid value across the board to determine the angle of the neighboring pixel this makes the algorithm lighter and the neighboring dip values usually does not differ much from each other.

Code snippet 3: dipStraightning.m

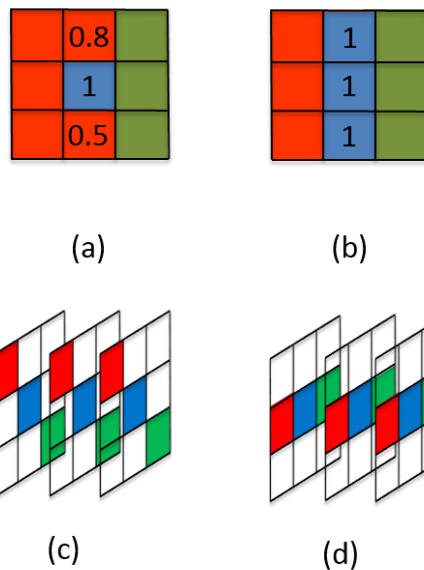
```

1 function output = dipStraightning(subCube,subXDipCube,subInDipCube)
2 output = subCube;
3 r = floor(length(subCube)/2);
4 k = r+1;
5 for j = 1:k
6     for i = 1:r
7 %     Check dip in in-line direction
8         if subInDipCube(k,k,k)<0
9             output(k,k+i,j)=(subCube(k+i,k+i,j)-subCube(k+i-1,k+i,j))...
10                *subInDipCube(k,k+i,j)+subCube(k,k+i,j);
11
12             output(k,k-i,j)=(subCube(k-i,k-i,j)-subCube(k-i+1,k-i,j))...
13                *subInDipCube(k,k-i,j)+subCube(k,k-i,j);
14         else
15             output(k,k+i,j)=(subCube(k-i,k+i,j)-subCube(k-i+1,k+i,j))...
16                *subInDipCube(k,k+i,j)+subCube(k,k-i,j);
17
18             output(k,k-i,j)=(subCube(k+i,k-i,j)-subCube(k+i-1,k-i,j))...
19                *subInDipCube(k,k-i,j)+subCube(k,k+i,j);
20         end
21     % do the same for cross-line cube
22     ...

```

The code snippet of dipStraightning (see Code snippet 3) shows only the code for the in-line direction. However, the process is identical for the cross-line direction using the second input; subXDipCube.

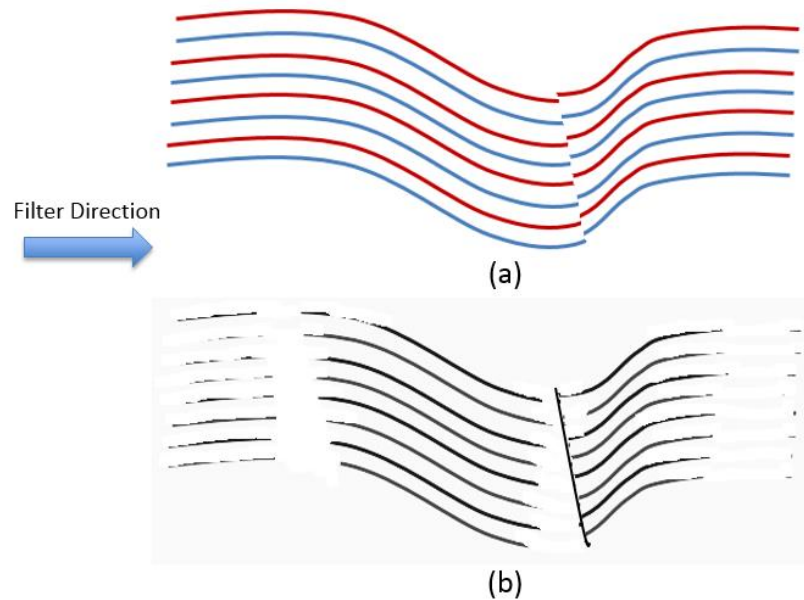
Figure 17 Equal dip straightening process



The reason we want to straighten the data before an edge detection is to avoid unnecessary edges to appear on the results. If we look at Figure 18, in (a) we see a representation of layered segments that we can see in a seismic dataset. From the left, we can see the segments starts out relatively flat before bending downwards. At the bottom of the slope, we can see the segments are offset a little upwards creating

a diagonal line; this is a fault we want to highlight. At the end, the segments slope upwards before straightening out to a flat surface.

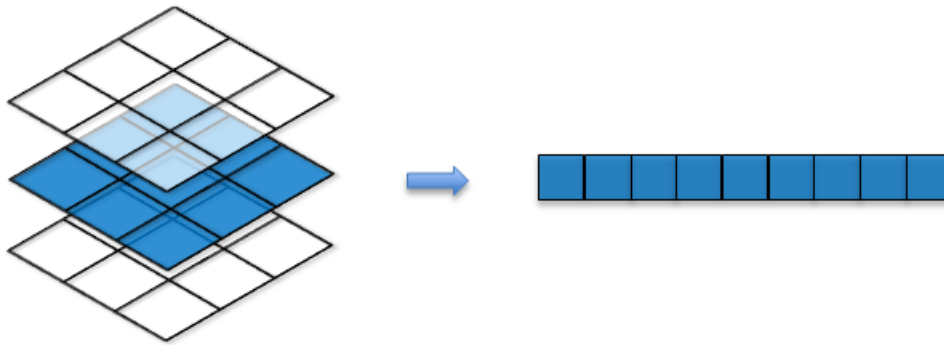
Figure 18 Dip straightening to avoid unwanted edges, (a) input, (b) output



Applying an edge detection along the filter direction, the results would look something like in Figure 18 (b), where the slopes along with the fault would be highlighted. If we were to use the edge detection results in further image analysis processes, the unwanted segments edges would pollute the end result when fault extraction is the goal.

4.4 VARIANCE ANALYSIS

With the newly dip-straightened sub cube from the previous operation, we send the data to the `varianceTest.m` function (see Code snippet 4). The test window in the time-slice plane is converted into an array (see Figure 19), as the variance function in MATLAB requires this, and normalized to achieve values between one and zero.

Figure 19 Time-slice plane converted into an array

The outputs of `varianceTest` are `noise` and `opSize`. The `noise` variable is the variance result of the input, while the `opSize` is the noise subtracted from the set value of 1.9 that we achieved best results. If the set value is more than two, the results becomes more blurry. If the set value is less than 1.9, the results becomes less detailed. The `noise` variable is used in the main algorithm to determine if the dip-guided data is to be used or discarded.

Code snippet 4: `varianceTest.m`

```

1 function [noise,opSize] = varianceTest(subcube,r)
2 myVector = reshape(subcube(r+1, :, :), [], 1);
3 myVector = myVector./max(abs(myVector));
4
5 noise = var(myVector);
6
7 opSize = 1.9-(noise);
8 if isnan(opSize)
9     opSize = 1;
10 end

```

With the normalized array, we find the variance according to Equation 7, and then we subtract this value from a set value to attain a new operator size for the next step. We also use the variance result to determine if the dip-straightened data not too noisy. If the noise is above a set threshold, we discard the dip-straightened data and use the original seismic data instead. This is because dip guiding does not perform well in chaotic areas such as salt domes.

Equation 7 Variance

$$\text{Var}(X) = E[(X - \mu)^2]$$

4.5 INTERPOLATION

Before using the Sobel operation, we need to make sure that the sub-matrix is a 3x3 matrix. If the operator size is more than one, then the decimal represents the percentage of the outer values used for interpolation (see Equation 8).

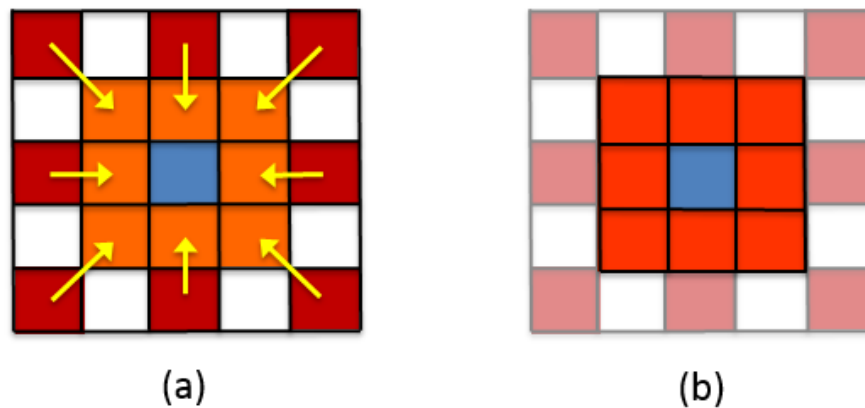
Equation 8 Interpolation

$$\text{Interpolated value} = x_1 * (1 - p) + x_2 * p$$

Where x_1 represents the value closest to the center, x_2 is the neighboring value to x_1 that is further away from the center. While p is the percentage of interpolation in the form of a decimal number between one and zero.

Looking at Figure 20 (a), we can see that some values are not considered for the interpolation process and taking the values that seems most natural in their position. The outer and inner border is combined to a set of values shown in (b).

Figure 20 Interpolation of a 5x5 sample to a 3x3 sample



In Code snippet 5, we can see the process of interpolating the values of an input matrix. At line 12, we create an empty two-dimensional matrix that will contain the interpolated values. The *int* variable is the floored value of the operator size that determines interpolation indexes, while the *dk* variable is the percentage of interpolation.

Code snippet 5: interpolationCube.m

```

1 function output = interpolationCube(subcube,opSize,center)
2 int = floor(opSize);
3 dk = opSize-int;
4 d = length(subcube);
5 center = ceil(d/2);
6 try
7     subC = reshape(subcube(center,:,:),[d,d]);
8 catch
9     subC = subcube;
10 end
11
12 sobelCube = zeros(3,3);
13 x1 = subC(center,center);
14 if center-int < 0
15     output = subC(:,:);
16 else
17     sobelCube(2,2) = x1;
18     try
19         sobelCube(1,1) = subC(center-int,center-int)*(1-dk)+...
20             subC(center-int+1,center-int+1)*dk;
21     catch
22         sobelCube(1,1) = 5;
23     end
24     sobelCube(1,2) = subC(center,center-int)*(1-dk)+...
25         (subC(center,center-int+1)*dk);
26     sobelCube(1,3) = subC(center+int,center-int)*(1-dk)+...
27         subC(center+int-1,center-int+1)*dk;
28     sobelCube(2,1) = subC(center-int,center)*(1-dk)+...
29         subC(center-int+1,center)*dk;
30     sobelCube(2,3) = subC(center+int,center)*(1-dk)+...
31         subC(center+int-1,center)*dk;
32     sobelCube(3,1) = subC(center-int,center+int)*(1-dk)+...
33         subC(center-int+1,center+int-1)*dk;
34     sobelCube(3,2) = subC(center,center+int)*(1-dk)+...
35         subC(center,center+int-1)*dk;
36     sobelCube(3,3) = subC(center+int,center+int)*(1-dk)+...
37         subC(center+int-1,center+int-1)*dk;
38     output = sobelCube;
39 end

```

4.6 SOBEL EDGE ENHANCEMENT

For this thesis, we are using a two dimensional Sobel filter on the time-slice plane and not three dimensional as used in [3]. Because complete straight lines is a rare occurrence in nature and it would seem that a three dimensional Sobel detriment more than it would benefit. In addition, a multitude of extra operations is avoided; such as dip straightening and Sobel filtering of multiple layers.

Figure 21 Sobel operators for x-axis (a) and y-axis (b)

1	0	-1
2	0	-2
1	0	-1

(a)

1	2	1
0	0	0
-1	-2	-1

(b)

By taking the square root of sum (a) and (b) in the second degree, we also get a smoothing (blur) across the image. If we look at Code snippet 6, we ensure that the input is a 3x3 square. In line 10 we can see ‘*2’ has been commented out; however, this can be included to strengthened the output value.

Code snippet 6: sobel.m

```

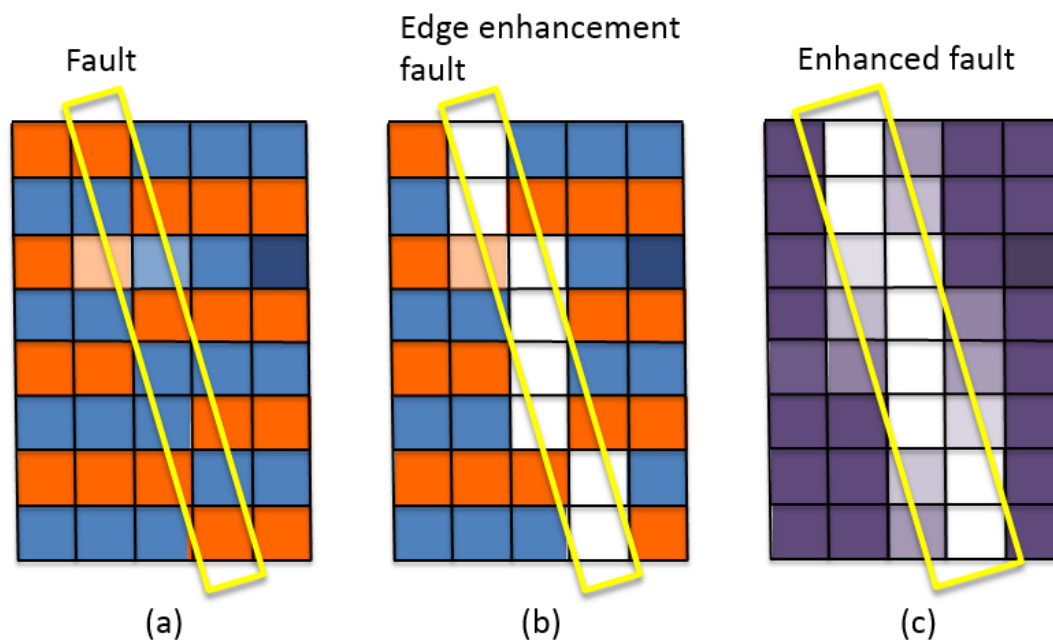
1 function v = sobel(subcube)
2 sx = [1 2 1; 0 0 0; -1 -2 -1];
3 sy = [1 0 -1; 2 0 -2; 1 0 -1];
4 r = floor(length(subcube)/2)+1;
5
6 tempSlice = reshape(subcube(r,r-1:r+1,r-1:r+1),3,3);
7 gx=sum(sum(tempSlice.*sx));
8 gy=sum(sum(tempSlice.*sy));
9 mySqrt=sqrt(gx*gx + gy*gy);
10 v=mySqrt;%*2;

```

4.7 VERTICAL SMOOTHING

After the attribute is finished with the Sobel operation on the desired time slice, a mean smoothing operation is performed in the vertical direction. The smoothing has a length of seventeen pixel, eight pixels above and below the current position. This is done to give emphasis to faults in the dataset while removing noise and shadows. The larger the filter size then more calculations is needed, if it is too small, fault continuity can be lost and more noise will be visible. In Figure 22, we illustrate how faults are highlighted with the use of vertical smoothing.

Figure 22 Vertical smoothing illustration



In Figure 22 (a), the different segments of the seismic are represented with the red and blue squares (positive and negative values). The faults can be seen where the squares change from one color to another in the horizontal plane as highlighted in (a). After an edge enhancement, it would look something like illustration (b). In the final illustration (c), we have applied vertical smoothing to remove noise and shadows making the red and blue squares into purple, and the fault have been widened making an enhanced fault.

Code snippet 7: verticalSmoothing.m

```

1 function output = verticalSmoothing(input,timeslice,r)
2 temp = input;
3
4 for i = 1:650
5     for j = 1:950
6         temp(timeslice,i,j) = mean(input(timeslice-r:timeslice+r,i,j));
7     end
8 end
9 output(timeslice, :, :) = temp(timeslice, :, :);

```

The vertical smoothing method is quite simple (see Code snippet 7). For each point in the time-slice plane, we find the mean value. This is the reason why the previous steps are done on eight time-slices above and below the one that are presented in the viewer.

CHAPTER 5

5 RESULTS AND ANALYSIS

In this chapter, we will introduce results for each segment of the algorithm and do some analysis and comparison to see how much an improvement each individual part contributes to the final result. This chapter will be divided into five sections.

In section 5.1, we will look at how the original seismic looks like and show how the Sobel operator performs on the seismic data without any prior or following enhancements.

In section 5.2, we will show Sobel performed on dip-straightened data and compare it with Sobel performed on the original seismic data. We will introduce two approaches of dip guiding; using the dip cube for all points, and an adaptive dip guiding if the point is too chaotic, the original seismic data are used instead of the dip straightened.

In section 5.3, we introduce interpolation of the seismic before performing Sobel filtering to see its effects on the results.

In section 5.4, we will show the effects of vertical smoothing after Sobel filtering and interpolation.

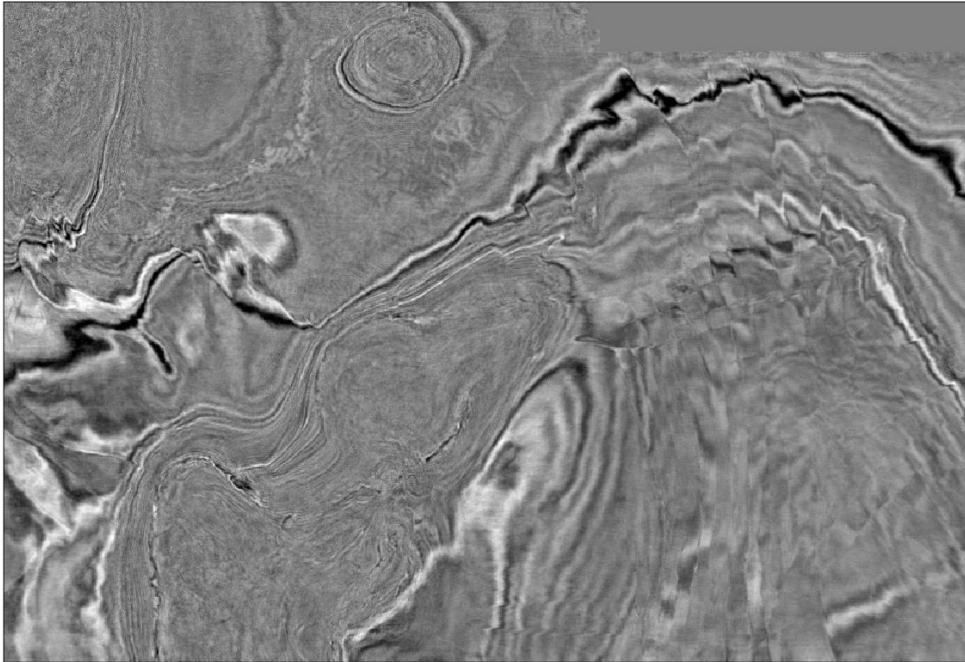
In section 5.5, we will compare our results with three-dimensional Sobel filtered output data from Petrel, which uses dip guiding and vertical smoothing.

In Section 5.6, we will discuss the inclusion of Gaussian smoothing before running the algorithm.

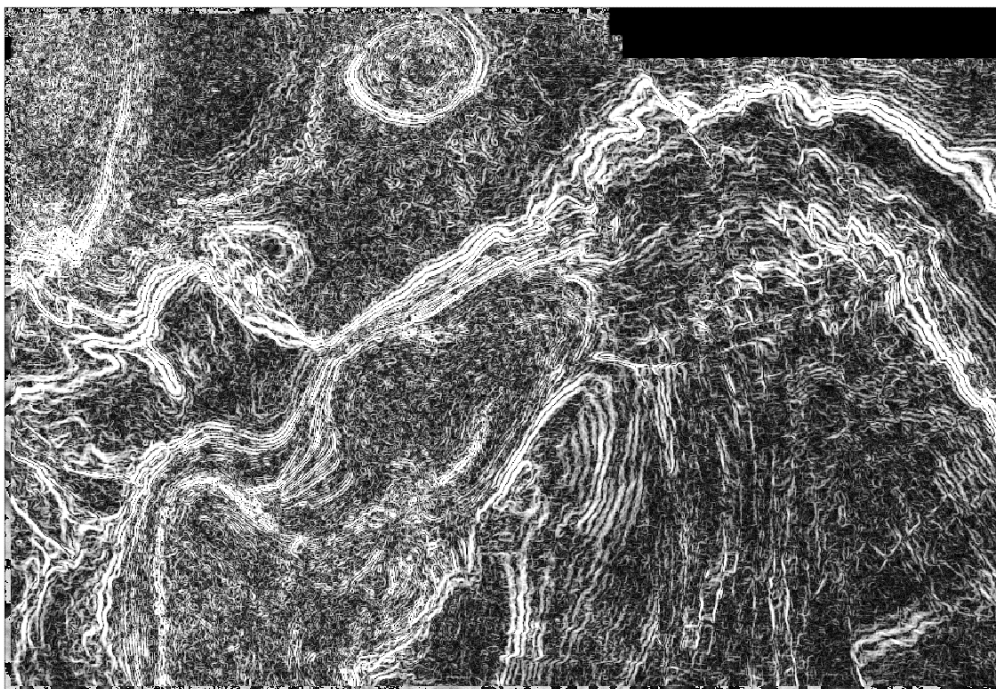
In section 5.7, we will look at the run time of the algorithm with the built-in feature in MATLAB.

5.1 SOBEL ON PURE SEISMIC DATA

Before looking at the Sobel results, let us get a view of the original seismic. We will focus on time-slice 1788 from the corresponding dataset F3 (447 in the seismic viewer). In Figure 23, we can see the seismic data unaltered, our main focus is the lower right area that are particularly faulty. After this section, we will focus on the area from 0 to 400 in the cross-line axis (denoted as xline) and 0 to 500 in the in-line axis, and do a side-by-side comparison and point out other differences throughout the other sections.

Figure 23 Original seismic

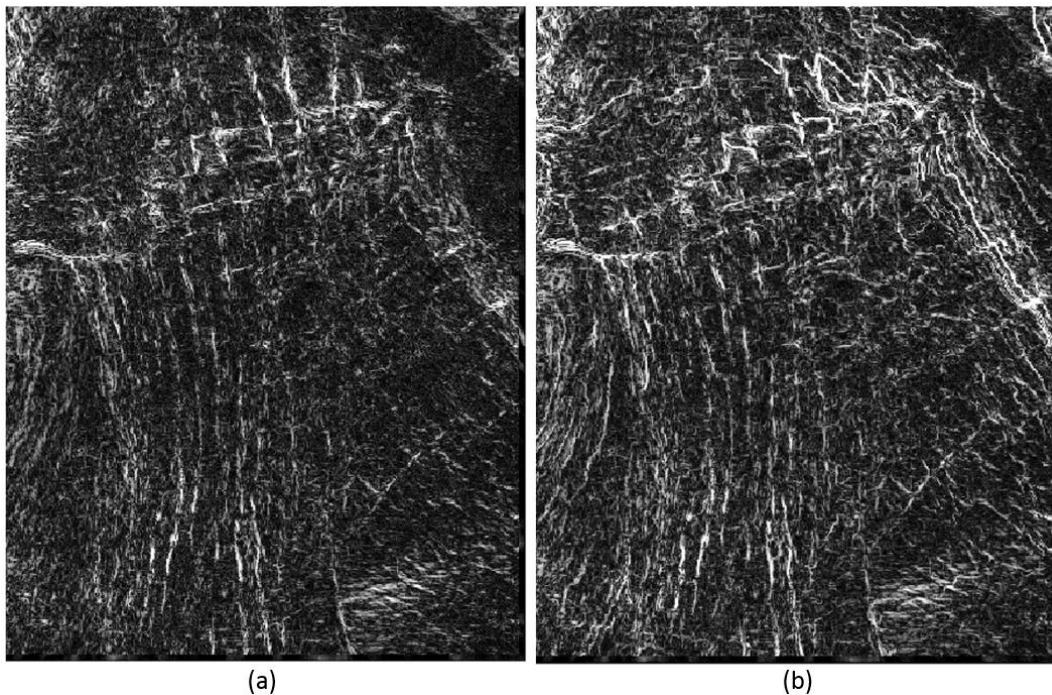
In Figure 24, we have applied a Sobel filter without any prior nor subsequent enhancement, and as one would expect; the Sobel operator picks up the stratigraphy (zebra stripes) easily, as illustrated with Figure 18. The F3 dataset is noisy and edges are present throughout the whole image, hence seeing the faults among the all the noise and stratigraphy are not easy.

Figure 24 Original seismic with Sobel filter

5.2 SOBEL WITH DIP GUIDING

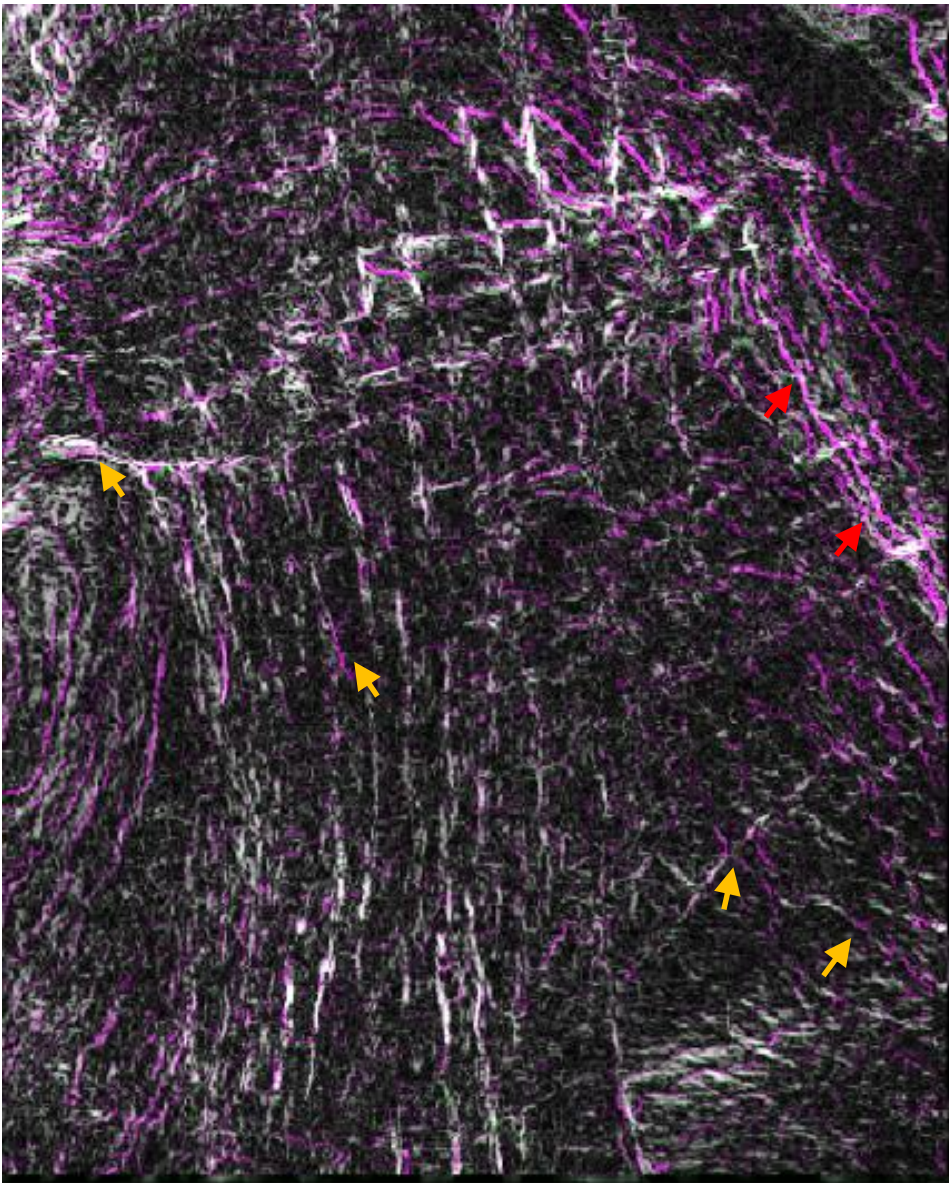
In Figure 25 (a), we can see an improvement of the results of Sobel filtering with the help of dip guiding. These dip cubes (in-line and cross-line) were created with the Petrel software. The stratigraphy are reduced, and we are starting to see some faults appearing from the faded stripes. However, it is not easy to determine faults from stratigraphy and noise at this point, but it is still an impressive improvement.

Figure 25 Sobel with dip guiding (a) and adaptive dip guiding and Sobel filter (b)



In Figure 25 (b), we can see that the stratigraphy are more apparent (present) with the adaptive dip guiding then with regular dip guiding. The faults are a little stronger than the previous image, although more noisy. These two approaches takes equal amount of processing power as the data are straightened out with dip guiding before a variance analysis is done. If the dip guided data is above the noise threshold the regular seismic data for the current sub-cube is used in Sobel filtration.

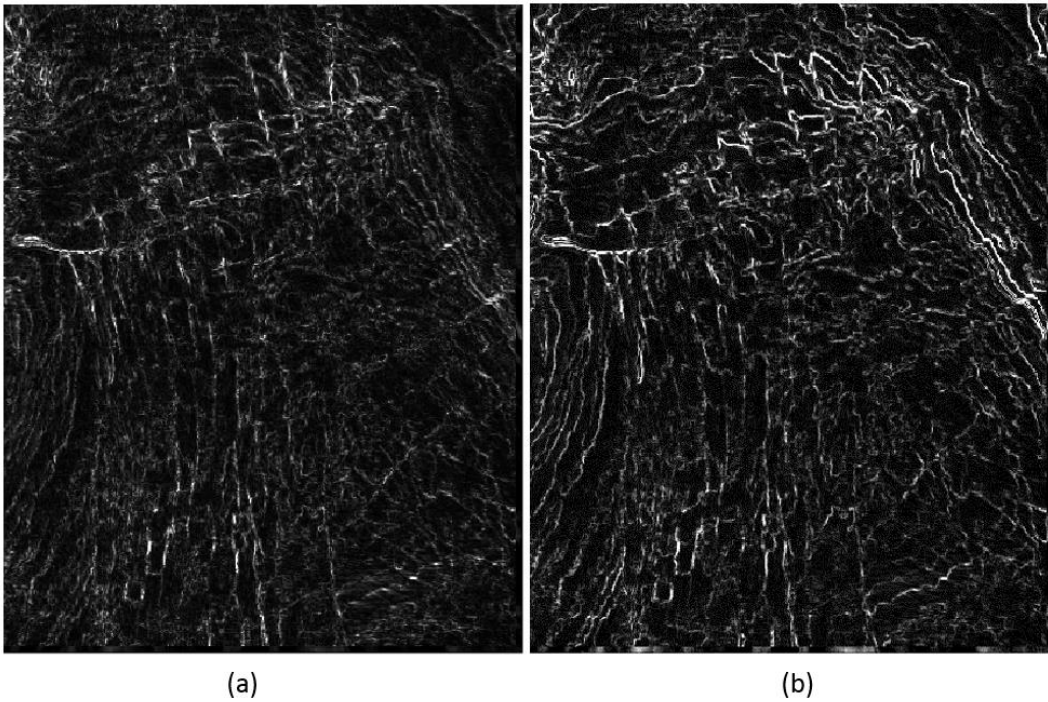
By doing a comparison of these two results, we get Figure 26. Here we can see clearly pink lines throughout the image, these pink lines represent what adaptive dip guiding has that regular dip guiding does not, while green represent what regular dip guiding has and adaptive does not. We have pointed out some faults that have been enhanced from using adaptive dip guiding with orange arrows and some less than ideal with red arrows.

Figure 26 Comparison of Comparison of results (regular dip guiding versus adaptive)

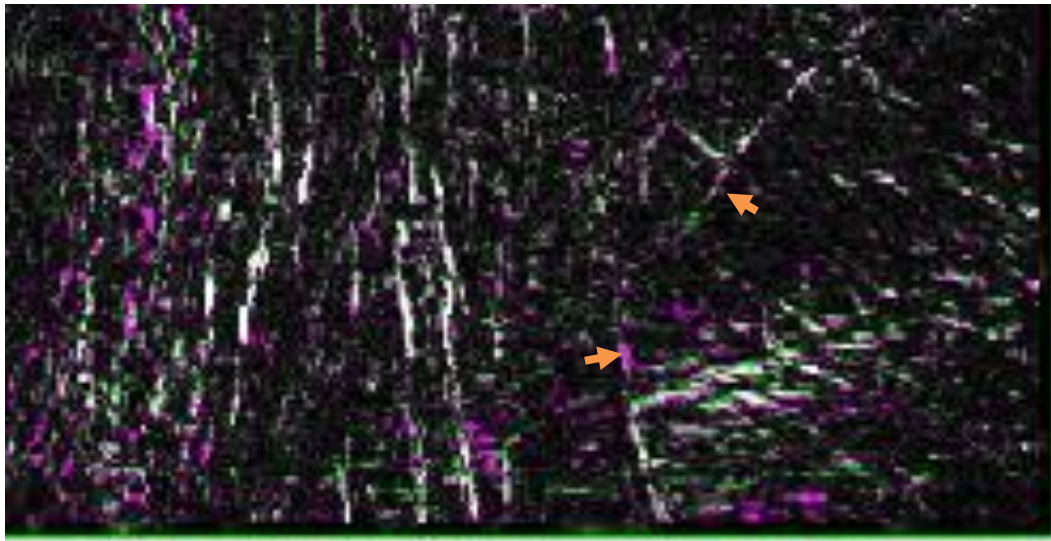
5.3 SOBEL WITH DIP GUIDING AND INTERPOLATION

After a variance test along the dip (after dip straightening), we determine a new operator size to use with interpolation including the noise variable. We continue to test on adaptive and non-adaptive dip guiding data. If we start looking at Figure 27 (a) with dip guiding and linear interpolation there are some small changes when comparing it to Figure 25. In Figure 28 (brightness, contrast and saturation have been increased to make it easier to see), we have highlighted (in purple) a couple additions that are not present in Figure 25 (we have concentrated in at the lower part of the image); these additions pointed out are faults that gives a better continuity.

Figure 27 Dip guided Sobel with interpolation (a) and adaptive dip guided Sobel with interpolation (b)

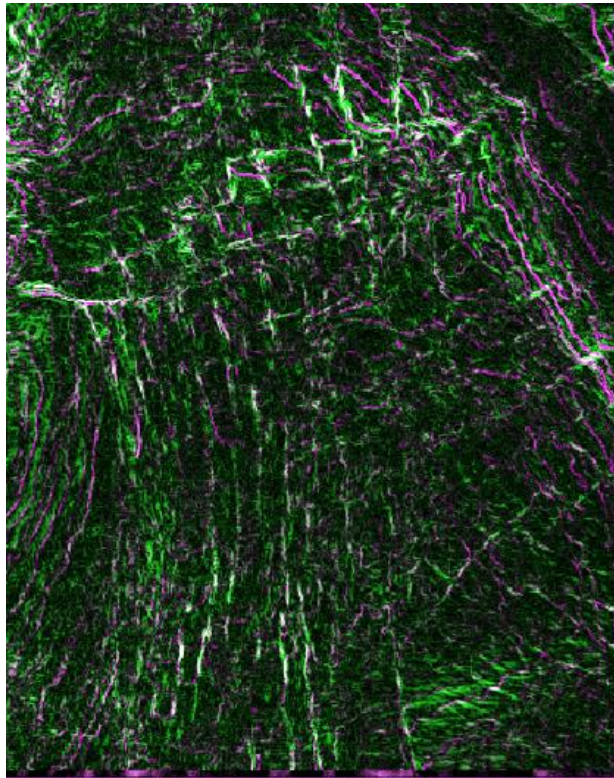


In Figure 27 (b), we can see the results of using adaptive dip guided Sobel with interpolation. With adaptive dip guiding, the interpolation results gives it a darker appearance with less sporadic noise, but also more of the stratigraphy is seen resembling Figure 27. This makes it difficult to distinguish faults from the crossing points of negative and positive values. This happens because of the use of original seismic data when deemed too noisy. This is also the reason the image gets overall darker, with dip guiding the interpolation happens with similar values (positive with positive and negative with negative), hence, without dip guiding, positive and negative values are interpolated giving less occurrences with values that differentiate; resulting in more lower values.

Figure 28 Difference between dip guided Sobel and dip guided Sobel with interpolation

If we look Figure 29, the green shows what dip guided Sobel has that adaptive does not, and vice versa with the pink. Here we can see that some parts of the faults are highlighted stronger by the regular dip guiding while the adaptive dip guiding highlights some other parts of the same faults. However, the regular dip guided image has a little more noise between the faults and the adaptive dip guided has stratigraphy.

Figure 29 Comparison of dip guided Sobel with interpolation and adaptive dip with interpolation

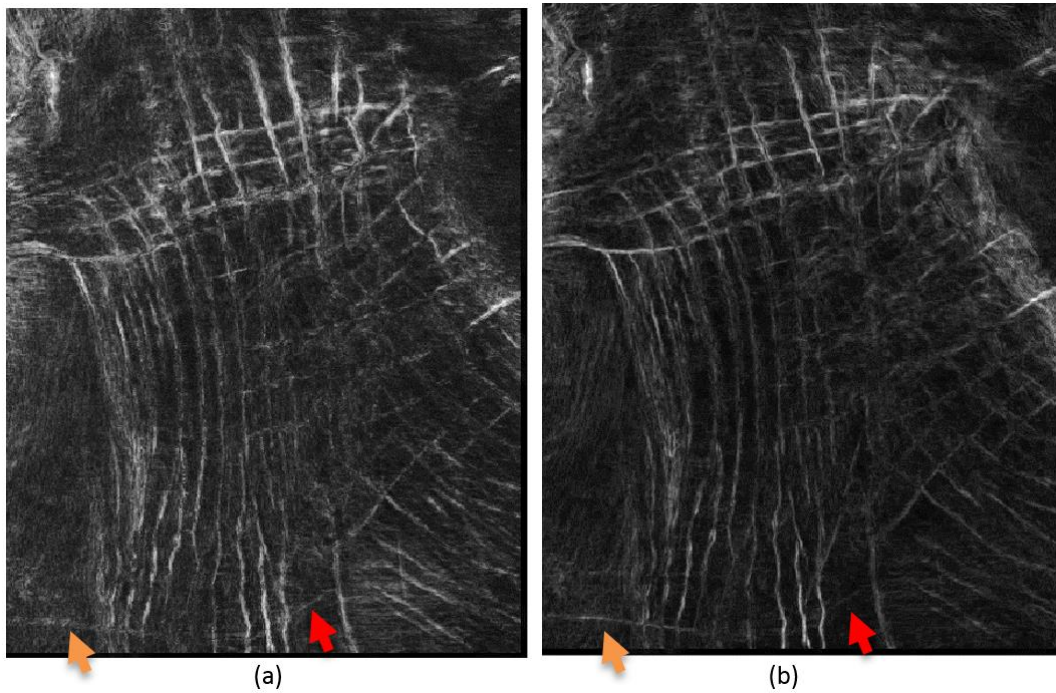


5.4 VERTICAL SMOOTHING

Before performing the vertical smoothing on the seismic cube, we do the dip guided Sobel filtering with interpolation on time-slices above and below the selected time-slice for viewing. Then by applying a vertical mean filter, we strengthen the fault visibility and weaken unwanted noise. In Figure 30, we can see the result of a vertical smoothing with a diameter of seventeen, eight above and eight below selected time-slice. We can now really see the faults pop out in the right part of the image. With the use of adaptive dip guiding, the image has a little darker tone. This is because of the combination of adaptive dip guiding and linear interpolation, where the use of original seismic instead of dip guided. The interpolation is likely to add a positive number with a negative one, hence, a low differentiation of the Sobel operator, giving close to zero as value.

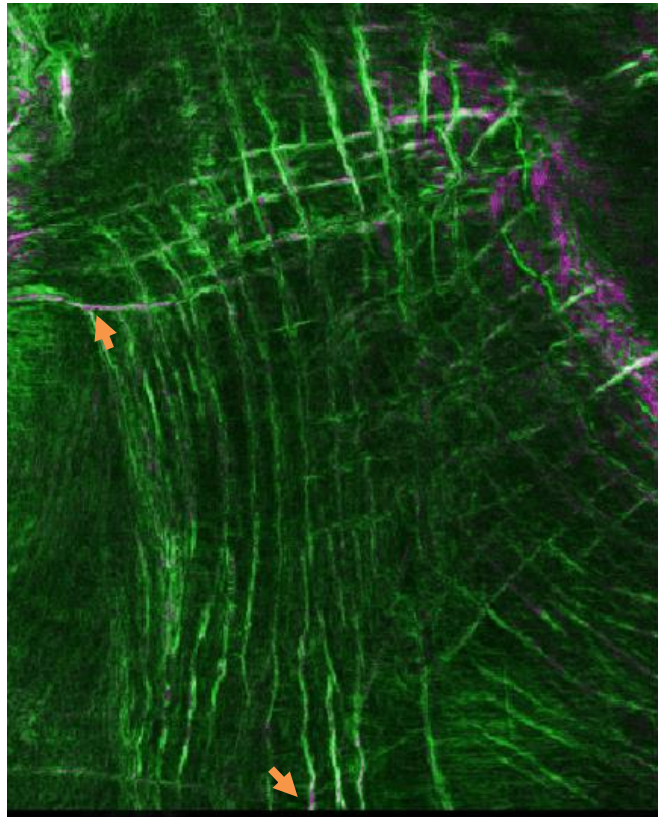
We have pointed out two differences between the images in Figure 30, the orange arrow shows how the adaptive dip guiding enhances this fault to a clear line, while in regular dip guiding, the line is a diffuse line that can be mistaken as noise. If we take our attention to the red arrows, we can see that this fault has higher value in regular dip guiding than in adaptive dip guiding

Figure 30 Dip guided Sobel with interpolation and vertical smoothing (a) and adaptive dip guided Sobel with interpolation and vertical smoothing (b)



By doing a comparison of these two results seen in Figure 30, we get Figure 31, It might seem discouraging too all the faults in green as it indicates a higher values in (a) than in (b). However, we can see that regular dip guiding produces more noise around and in between the faults except up in the right area. We have marked two areas (orange arrows) where the adaptive dip guiding gives stronger results than with regular dip guiding.

Figure 31 Comparison of regular dip and adaptive dip with interpolation



The mean smoothing operation in itself does not take a lot of time. However, it is time consuming to do the fast Fourier transform, dip straightening, variance analysis, linear interpolation and Sobel operation for all the time-slice layers needed. Nonetheless, comparing it with the results of Sobel filter with dip guiding on one time-slice layer (see Figure 25), the amount of details, noise removed and continuity achieved is staggering. The added complexity of the algorithm is worth the time in this case, as the code used can be optimized greatly.

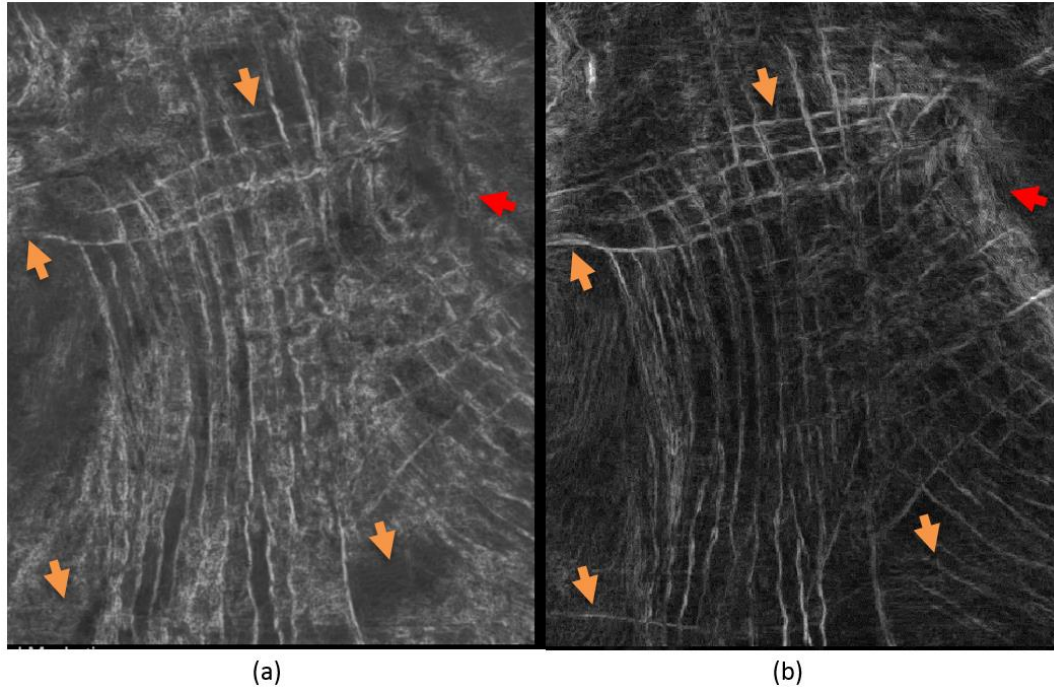
5.5 COMPARISON WITH PETREL RESULTS

As mention in Chapter 2.3 Petrel, we want to do a comparison of our results with how Petrel on the same dataset. The Petrel attribute is a three-dimensional Sobel filter with dip guiding and vertical smoothing and one including a 5x5 median filter along the dip as well. For the comparison we have used the adaptive dip guided Sobel filtered with vertical smoothing results. We have also adjusted the brightness and contrast of our result to be more easily comparable with the Petrel results.

In Figure 32, we have pointed out some of the differences between the Petrel results and our results. The orange arrows show that our approach does better to extract faults, the continuity are better for the most part and faults that are not visible in Petrel, pointed out by the two lowest arrows. The red arrow points out that our

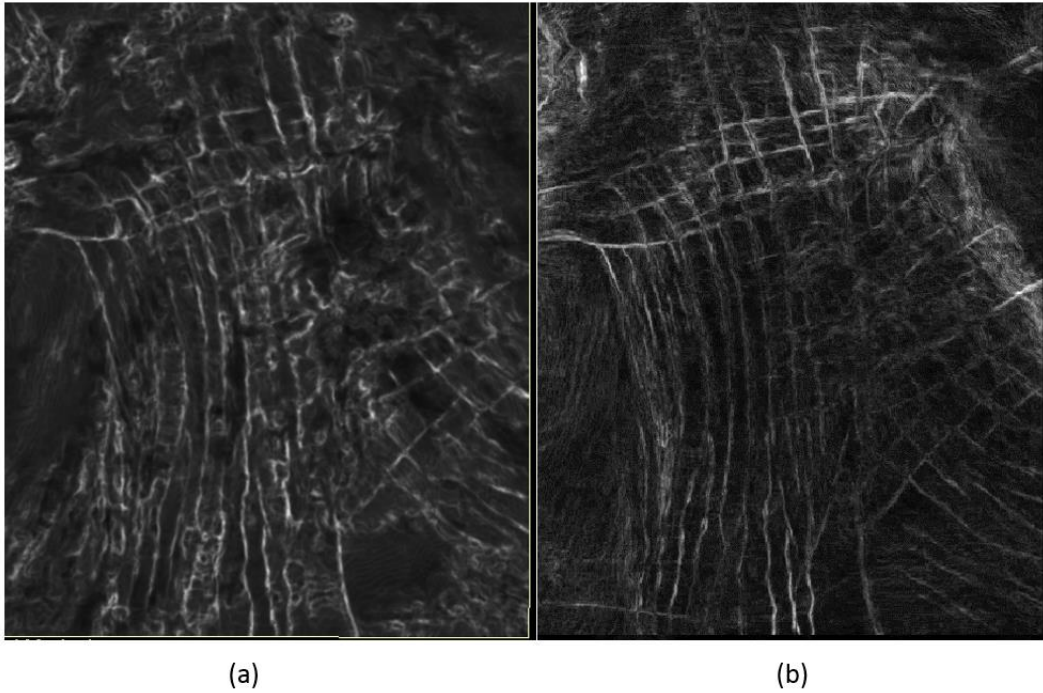
approach does feature shadows produced by the stratigraphy captured by the adaptive dip guiding. However, inside this shadow, several faults are present that is not captured by the Petrel algorithm. After increasing the brightness and decreasing contrast on our result, we can see more of the Gaussian noise present in the dataset. On the other hand, the noise is stronger in (a) that makes it harder to determine the continuity of faults on a computer imaging basis.

Figure 32 Dip guided three-dimensional Sobel filter (a) and our adaptive dip guided Sobel with interpolation (b)

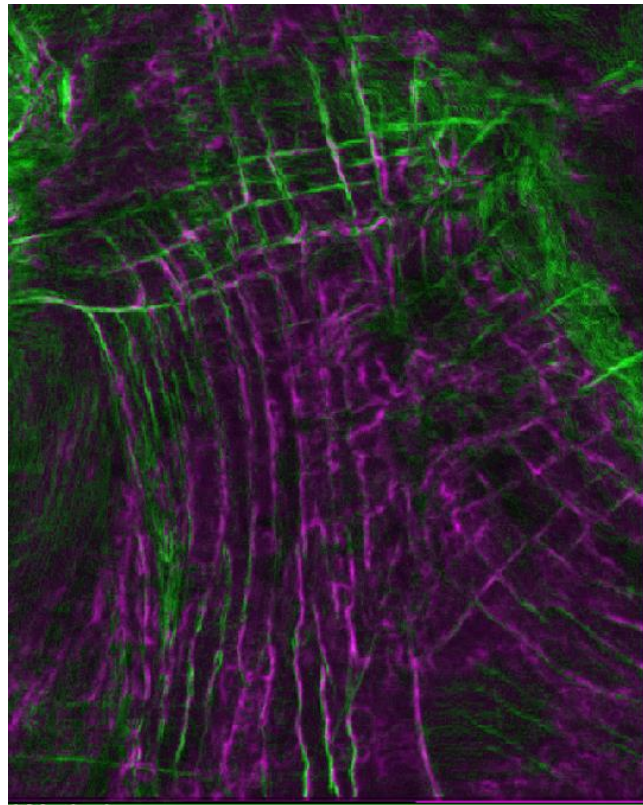


If we look at (a), a 5x5 median filter has been applied to reduce noise and it shows to be very effective. A lot of noise between faults has been removed and faults are more easily distinguishable. Nevertheless, our results show more information regarding faults and the noise present does not have high values that diminishes faults.

Figure 33 Dip guided three-dimensional Sobel filter with 5x5 median on top (a) and our adaptive dip guided Sobel with interpolation (b)



In Figure 34, we can see a comparison between our and Petrels result, the green indicates where our values are higher and pink indicates higher values for Petrel. The faults in the middle of the image and downwards, Petrel has higher values on the faults. However, we want to remark that the noise in (a) overshadow weaker faults in (b).

Figure 34 Comparison between our results and Petrel results from Figure 33

5.6 GAUSSIAN SMOOTHING PRE-PROCESSING

We applied a 3x3 and 5x5 Gaussian-smoothing filter, to see if it would give an improvement of the Sobel operation. This however, proved to be fruitless, as it had no visual effect after running our algorithm. With the included expense in time added to the algorithm, it will not be included as a part of the final algorithm.

5.7 MATLAB ALGORITHM RUN TIME

MATLAB has a feature where we can time the algorithm and see how much time each process takes. In

Table 1, we can see the procedures sorted after total time taken from highest to lowest. The algorithm traverses $N \times M$ where $N \leq M$ points and seventeen in the third dimension, this gives it a complexity of $O(N^2)$.

The function on top, runTimeTest, is the function that calls the algorithm; timeSliceAttribute, which runs all the functions. We can see that createOperatorSizeCube function uses a two thirds of the total time, because it calls opSizeSelector and hamming. Because fast Fourier transform is in the signal toolbox of MATLAB, it calls all “signal/private/...” functions. After this the variance testing uses 443 second that includes the var function that lies beneath it.

5 Results and Analysis

The interpolation function uses 235 seconds and uses no extra calls, while Sobel filtering uses 136 seconds in total.

The vertical smoothing with mean filter uses the least amount of time, clocking in close to eighteen seconds.

Table 1 MATLAB algorithm run time

Profile Summary

Generated 30-May-2015 16:29:51 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
runTimeTest	1	2860.569 s	0.217 s	
timeSliceAttribute	1	2858.992 s	241.376 s	
createOperatorSizeCube	1	1629.884 s	56.825 s	
opSizeSelector	12350000	1573.059 s	164.778 s	
hamming	12350000	1408.281 s	116.394 s	
signal\private\gencoswin	12350000	1291.887 s	343.985 s	
varianceTest	10200357	443.126 s	142.108 s	
cell_strmatch	12350000	338.157 s	163.130 s	
var	10200357	301.018 s	301.018 s	
sigcasttfloat	12350000	262.443 s	128.060 s	
interpolationCube	10200357	235.573 s	235.573 s	
signal\private\gencoswin>sym_window	12350000	212.658 s	107.069 s	
strmatch	12350000	175.027 s	175.027 s	
dipStraightning	10200357	155.202 s	155.202 s	
sobel	10200357	136.080 s	136.080 s	
signal\private\check_order	12350000	134.644 s	134.644 s	
sigcheckfloattype	12350000	134.383 s	134.383 s	
signal\private\gencoswin>calc_window	12350000	105.589 s	105.589 s	
verticalSmoothing	1	17.751 s	8.835 s	
mean	617500	8.916 s	8.916 s	

CHAPTER 6

6 CONCLUSION AND FUTURE WORK

Exploring the subsurface through seismic imaging has been, and will continue to be determined by educated guesses. However, with the increasing endeavors in image processing, these educated guesses become more accurate.

We will conclude this thesis with two sections. In section 6.1, we will summarize our results and the impact on the dataset. Lastly, section 6.2, we will discuss possible future work that can be done within this field.

6.1 CONCLUSION

The approach of an adaptive operator size for a Sobel based edge detection to achieve a more detailed image proves to be successful as it surpasses the industry standard for fault detection set by the Petrel Platform. Some of the larger faults in our test data are weaker in value than in Petrel, however, the noise in between the faults has been reduced substantially and smaller faults parallel and perpendicular have emerged. This in return gives a more complete and realistic fault network, which is a much better starting point for any interpretation workflow. This way risk is reduced and simulation models can be more complete.

The discrete Fourier transform approach to determine an operator size is time consuming while its impact on the result is minimal. The key advantages are of course the more accurate estimation. However, through the use of other frequency or chaos measurements we are able to achieve similar results with a fraction of the computation cost. Hence, we conclude that the Fourier transform method in this case is not ideal.

With the help of the dip cubes that were created using Petrel, significant improvements were made in applying the Sobel operator. It allows us to follow the curvature of the seismic layers for the most parts, avoiding zero crossings to make an impact on the fault detection process. Ultimately, this reduces the capture of geological noise. Using our conditional logic of using the dip guiding only when in less chaotic areas, we see a clear improvement in the continuity of the faults detected. This is due to the fact that most dip estimations are unstable in highly chaotic regions. We achieved this by using variance testing along the dip to check whether or not it is too chaotic. This has improved the detection of weaker faults that would otherwise be excluded in the dip guiding process. This also determines a new operator size to be used on the interpolation function, giving a seemingly higher resolution to the fault network

6.2 FUTURE WORK

Here we present some suggestions for future work within the research field:

Improved frequency analysis to determine starting operator size, explore new frequency detection methods. Because the chaotic nature sub-surface, frequency changes not only by depth, but also after what material the layers are made of. Hence, to focus on the dominant frequencies that surround the filter operation site would help finding the best matching operator size for the current situation. This would account for acquisition noise to correct frequency detection.

Improved interpolation model: the interpolation model used is quite simple, it does not require a lot of processing. However, because seismic is received by sound it is natural to assume a cardinal sine (sinc) interpolation might be the better choice. This would however increase the algorithm cost. Another possibility is to limit the interpolation for values far from the center. When the operator size is small, interpolating values might not be needed and if the operator size is large, interpolating the outer values could be too simple of an approach.

Try different edge detection methods: in this thesis Sobel was the only approach used because of its cost effectiveness. Other edge detections could be implemented, such as variance. With variance, edges caused by chaotic noise can be avoided [29]. Other weighting than Sobel approach for Gaussian derivative could be implemented. Hough transform, a feature extraction technique that aims to find imperfect instances of objects within a certain class of shapes through a voting procedure. It could also be used to find faults in a three-dimensional planes directly.

APPENDIX

APPENDIX A

The MATLAB source code is included as attachment. The structure is as follows:

- **Analysis**
 - **opSizeSelector.m** – FFT analysis
 - **varianceTest.m** – variance analysis
- **crewes** – *needed to read *.seggy files, please download from crewes.org website* [16]
- **cube_operations**
 - **convert2CubeForm.m** – converts the loaded seggy file into a three-dimensional matrix that are used for further operations
 - **createOperatorSizeCube.m** – runs opSizeSelector for each point in cube to create an operator size that are used later
 - **dipStraighting.m** – takes a sub-cube and flattened it out
 - **interpolationCube.m** – takes a sub-square and interpolate values to form a 3x3 before the Sobel operation
- **filters**
 - **sobel.m** – applies a Sobel operator and returns the value for a single point
 - **timeSliceAttribute.m** – this runs the algorithm on the selected time-slice
 - **verticalSmoothing.m** – applies a vertical mean smoothing on the selected time-slice
- **myUI**
 - **drawSlice.m** – takes the values of the cube and converts them into a RGB image
 - **mySeismicUI.m** – this makes the window with buttons and sliders for interactions
 - **plotInLine.m** – creates a surface with the dimensions of an in-line slice of the cube with the image produces by drawSlice.m
 - **plotTimeLine.m** – does the same as plotInLine.m for time-slice
 - **plotXLine.m** – does the same as plotInLine.m for cross-line
- **read_seggy_file.m** – contains the steps of reading a seggy file, converting it to a matrix and opening the seismic viewer. The seggy folder is needed

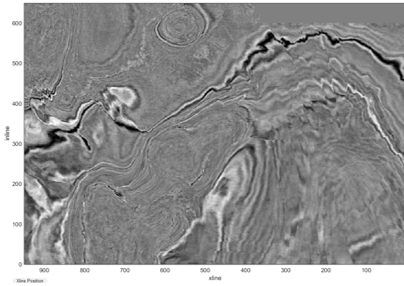
APPENDIX B

We have included a presentation poster that was a part of the Master thesis; it is split into four presentation slides.

Sobel Based Edge Detection Algorithm with Adaptive

University of Stavanger - Department of Electrical and Computer Engineering

Introduction

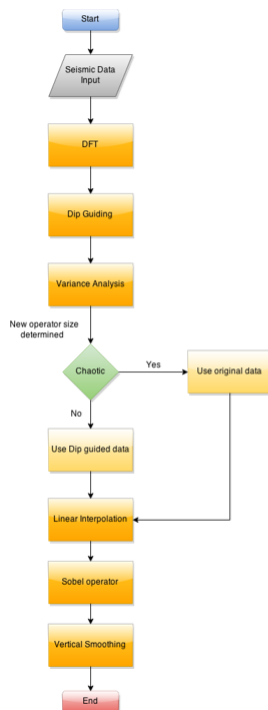


Seismic time-slice image 1788 milliseconds beneath the North Sea in the Netherlands Coordinates: N 54°52'0.86" / E 4°48'47.07 F3 data set

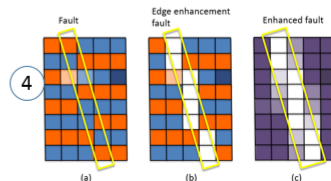
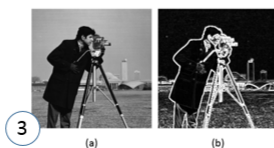
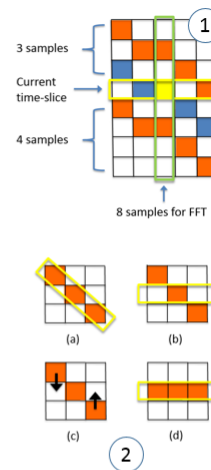
Exploring subsurface areas is mostly done by researching seismic imaging thoroughly before attempting to drill. To achieve the most information from seismic imaging, different types of image filtering are applied; which are called seismic attributes in the industry.

In this work, we want to adjust the operator size according to the frequency that is the most dominant around the filter operation site. The essence is to challenge the common approach the oil and gas industry.

Methods



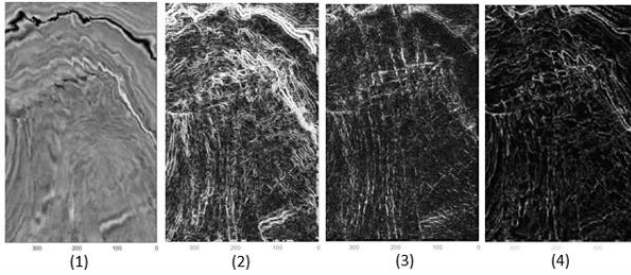
On the input seismic data we perform a discrete Fourier transform (using fast Fourier transform) on each pixel where we take three samples above and four samples below to get eight samples needed for a FFT (see figure 1). This determines an operator size to be used for the dip guiding process (see figure 2), this process straightens the data (c) and (d), instead of trying to apply filter along the appropriate value (a) or going across the values (b). We test the variance of the straightened data, if variance is too high we use the original data, if not we continue with the straightened. We then apply a Sobel operator (see figure 3) to detect edges, this is done for eight layers above and below the selected layer we want to view. Lastly we apply a vertical mean filter that gives the effect seen in figure 4 (c).



Operator Size Applied to Post-Stack Seismic Data

-David Sundli Barka

Results



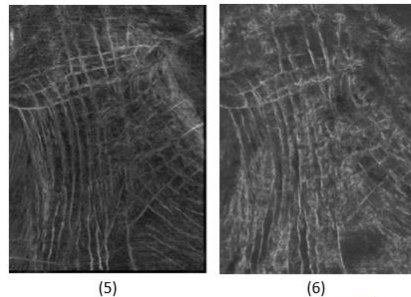
From left to right:

1. Original seismic data
2. Sobel filter applied on original data
3. Sobel filter on dip guided data
4. Sobel filter on adaptive dip guided data with linear interpolation

The area showed in the figures are heavily faulted (cracks), we can see what each part of the algorithm adds to the results. By adding Sobel directly on the seismic we highlight all the stratigraphy that takes the focus away from the faults. With dip guiding we filter along the segments, avoiding much of the stratigraphy. With linear interpolation we define a filter window that is not determined by pixels.

After applying the Sobel operator on the dip guided and interpolated data, we apply a vertical mean smoothing filter to increase fault visibility and reduce unwanted noise.

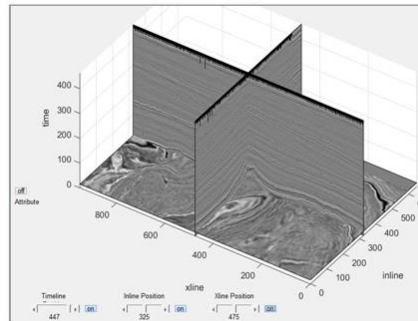
In figure 5 and 6, we can see a comparison of our result (5) and the industry leading solution of the Petrel Platform (6). We have removed noise between faults and increased continuity.



Conclusion



The approach of a Sobel based edge detection with adaptive operator size has proven to give a more detailed view of faults in a three-dimensional seismic. The faults have better continuity and are more isolated from unwanted noise and shadow that makes it more practical for 3D modeling and further attribute enhancements. This method could be proven useful for other attributes other than edge enhancement.





IPTC 17543

Adaptive Sobel Based Edge Detection for Enhanced Fault Segmentation

Ahmed Adnan Aqrabi, Schlumberger

Copyright 2014, International Petroleum Technology Conference

This paper was prepared for presentation at the International Petroleum Technology Conference held in Doha, Qatar, 20–22 January 2014.

This paper was selected for presentation by an IPTC Programme Committee following review of information contained in an abstract submitted by the author(s). Contents of the paper, as presented, have not been reviewed by the International Petroleum Technology Conference and are subject to correction by the author(s). The material, as presented, does not necessarily reflect any position of the International Petroleum Technology Conference, its officers, or members. Papers presented at IPTC are subject to publication review by Sponsor Society Committees of IPTC. Electronic reproduction, distribution, or storage of any part of this paper for commercial purposes without the written consent of the International Petroleum Technology Conference is prohibited. Permission to reproduce in print is restricted to an abstract of not more than 300 words; illustrations may not be copied. The abstract must contain conspicuous acknowledgment of where and by whom the paper was presented. Write Librarian, IPTC, P.O. Box 833836, Richardson, TX 75083-3836, U.S.A., fax +1-972-952-9435

Abstract

In this work, we suggest an improved method of traversing post-stack seismic data with an adaptive operator size for detecting edges. Attributes, in the coherence family, used for edge detection usually have a static operator size (Marfurt and Chopra, Seismic attributes for prospect identification and reservoir characterization, 2007). This can prove to be limiting in that one would under or over sample in regions of higher or lower frequencies respectively. By looking at the nature of seismic data, which commonly changes from short wavelength to longer wavelength signals in depth/time, an adaptive approach is more suitable for detection of features. By adjusting the operator size in relation to the depth/time, we are better following the frequencies of the data as we filter them. To increase the accuracy and resolution of our filtering we have chosen to interpolate between adjacent seismic traces. However, seismic data is not that simple in structure such that one can only vary in depth/time. Geological features such as dipping, salt and gas result in chaotic and varying frequencies regardless of when they occur. This is why we have introduced a textural analysis to account for this change and adapt our filtering to it. We have chosen to use chaos (Iske and Randen, 2005) as our seismic texture change indicator, as we are looking for changes to higher frequencies that usually results in chaotic textures.

The attribute we have chosen to implement is a 3D Sobel based edge detector, namely amplitude contrast (Aqrabi & Boe, 2011). In essence, the textural analysis in this case will decide the outcome of two things. One, the choice of preconditioning prior to filtering, and the second is the normalization method used. While, when implementing the operator size of our edge filter, relative depth/time is used to adjust it adaptively. As such, the algorithm operator size is time variant and accounts for seismic texture by varying the selected algorithm to filter with and the appropriate operator size to do so. To test our algorithm, a heavily faulted seismic dataset from the Norwegian North Sea has been used to test the methodology of the adaptive calculations. Our results indicate that the adaptive edge method ensures a higher level of detail, and highlights the smaller amplitude discontinuities better than a static operator size approach. It also proves to increase continuity and reduces the detection of noise, which overall gives a more accurate edge detection of the seismic data.

Adapting Attribute Calculations to the Nature of Seismic Data

Seismic data is highly varying in nature given the way it is acquired. There is more detail in the seismic image in the overburden where one has higher frequency information. As we propagate along time, we see that the loss of higher frequency signal gives longer wavelength information and hence is not as rich in detail. As such, similar geological features will look and respond differently depending on when they are captured. As an example, a fault is often seen as an amplitude discontinuity in higher frequency information, but a change in phase in lower frequency signals. In this example, we are looking at the same fault feature, but its representation in the seismic data has changed (see Figure 1).

Not only do we see a variation in the seismic signal in time, but also in seismic texture given changes in geological features captured (see Figure 1). For example, a chaotic texture for a geological feature such as salt will result in a noisier signal and hence capture higher frequency information than the surrounding signal data. This means that seismic data varies in time, but also along the lateral directions at the same time as we have variations in seismic texture. This is why it is more accurate to have a changing operator size that accounts for both variations along the vertical axis of the seismic data, as well as seismic texture. This basically means if one has high frequency information, using a smaller operator size will account for all the detailed features. While in low frequency regions, one uses a larger operator to avoid the detection of noise together with features

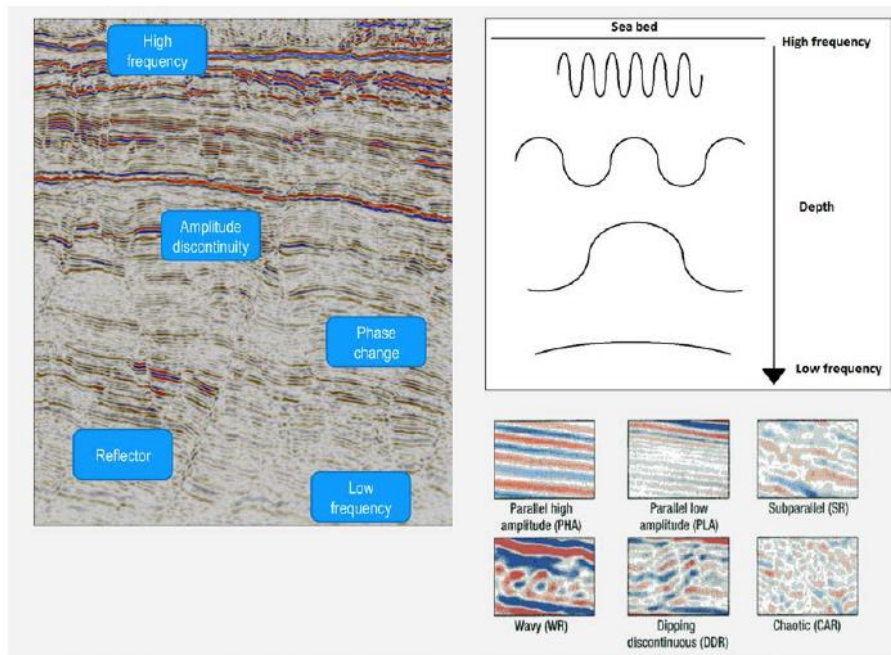


Figure 1: Here all the elements accounted for in the adaptive filter are shown to highlight the challenges faced when filtering seismic data. A seismic section is chosen to show the various changes occurring in seismic with time and illustration of the changes in frequency is put adjacent to it. Also shown are the variations of seismic textures taken from (Sonneland et. al, 2005), here the textures are ranged from the least chaotic (upper left), to the most chaotic (bottom right).

Understanding Faults in Post-Stack Seismic Data

As mentioned earlier geological features change imaging properties depending on when they are captured, and faults are no exception. Faults in post stack seismic data can be categorized to have three different properties in the image: amplitude discontinuity, change in phase, chaotic texture. As such one would need to have accurate measurements of all three components and combine them to get a clear segmentation (Aarre et. al 2010).

Fault picking is a manual and time consuming process on a slice by slice basis and one fault at the time, which requires very well trained interpreters. In the industry there are some automated approaches based on the premise that seismic faulting introduce discontinuities in the seismic horizons. In order to better define this discontinuity several seismic attributes have been shown to help like chaos (Iske and Randen, 2005), coherence (Bahorich and Farmer, 1995), variance (van Bommel and Pepper, 2000), curvature (Roberts 2001, Boe and Daber, 2010) and Sobel filter attributes among others (Chopra and Marfurt, 2008). Subsequently it is possible to extract the faults, and compare them to the original seismic for quality control.

With a classical approach to calculating seismic attributes, with the use of a constant operator size, one is going to either over or under sample during the calculations. This results in the detection of noise and ignoring of some features as they are skipped in sampling. In our work, we will focus on the detection of amplitude discontinuities, which is one of the three measurements needed to map faults. This will be done through the use of a 3D Sobel filtering technique discussed in (Aqrabi and Boe, 2011).

The Sobel filter is a computation of the amplitude derivatives between neighboring traces where the non diagonal neighbors are weighted twice as much. The calculated differences are then normalized and the final value is calculated using Equations (1), (2), (3) and (4), where S_x , S_y and S_z are the Sobel operators in the corresponding dimensions. Here the values are squared to avoid negative differences, and finally one takes the square root of the sum of the squared values to calculate the result.

In addition to the classical Sobel, we have added amplitude normalization and dimension weighting. The normalization helps to avoid discrimination between high and low amplitudes, and the weighting helps to minimize the horizontal artifacts of the vertical differences. Prior to the Sobel calculation the structural dip is estimated by finding the dip that maximizes trace-to-

trace cross correlation. Subsequently, amplitude values are extracted from the seismic volume parallel to the dipping plane. This helps in detecting discontinuities in the presence of dip.

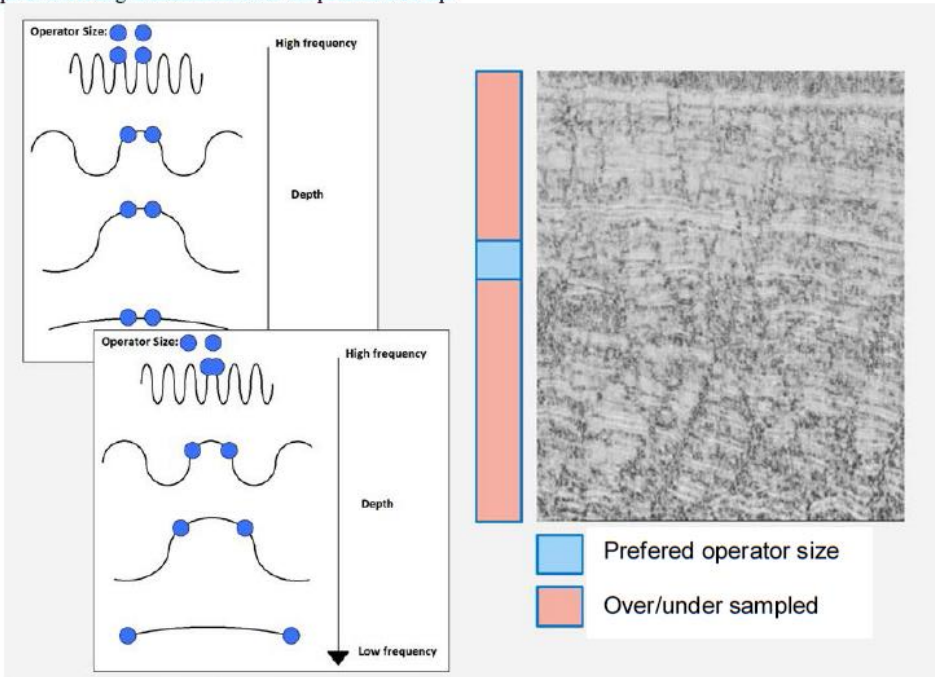


Figure 2: illustration of the effects of the over and undersampling on a filtered seismic section using Amplitude Contrast, when using a static operator size. The image also describes the nature of an adaptive filter versus a traditional one, with how the signals frequency content is accounted for. On the seismic section the effects of the preferred operator size versus the effects of over and undersampling are highlighted using blue and red color respectively.

With the use of the adaptive filtering technique (see Figure 2), we aim at reducing the detection of noise, the avoidance of over and under sampling when performing the calculations, and a more accurate measurement of amplitude discontinuity in post-stack seismic data.

The Adaptive 3D Sobel Filter

The traditional Sobel filter calculates changes in contrast between neighboring amplitude values. As such one selects the preferred operator size and performs the calculations on the samples made available. The implementation used to compute our adaptive filtering technique on, and to compare our results to, is the one in (Aqrabi and Boe, 2011). In this implementation, a modification has been done such that the calculation is dip guided, normalized and the various dimensions are weighted differently.

In our adaptive approach, we again use the same calculation, but we modify the operator size dependent on where in the seismic data one is performing the calculations. Our method is performed as follows:

1. Select an operator size given the position in time (this will grow linearly to account for loss of high frequency information)
2. Using this operator size, calculate Chaos (Iske and Randen, 2005)
3. Given the chaos results rescale the operator size to account for that chaotic textures will require smaller operator sizes even at a greater time. Chaos returns a value between zero and one, and gives an indication of how chaotic a seismic texture is.
4. Use the newly scaled operator size, that has now adapted its size to the seismic data, and calculate the 3D Sobel attribute.

For a graphical view of the methodology steps, and a further explanations see Figure 3.

As we scale the operator sizes, not all samples within the operator are accounted for. Rather we use a Sinc interpolation to get the values preferred and do the calculation on them. This is to enhance computation time.

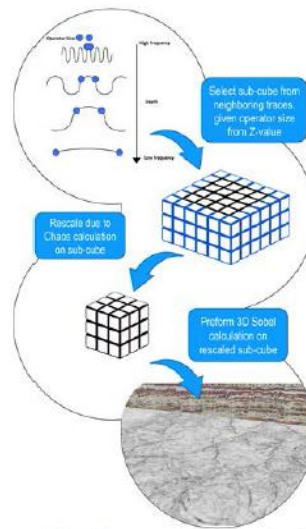


Figure 3: The adaptive edge detection attribute workflow illustrated to show what the various steps in the algorithm are. The workflow starts from top to bottom in the image, and explains the steps of adapting to the time, then adapting to the texture, to finally extrapolate the values and perform the calculation.

Test Case Data

The data set used in testing is Solsikke, and is from the Norwegian continental shelf (See Figure 4). The Solsikke structure is situated in the north-western part of the outer Møre Basin, close to the Møre Marginal High. The structure is defined as a large Tertiary dome comprising a thickness anomaly of Upper Cretaceous sediments (Norwegian Petroleum Directorate, 2011). On this data set we noticed challenges to use state of the art seismic attributes to extract faults, and aim to overcome these challenges with our adaptive Sobel based attribute.

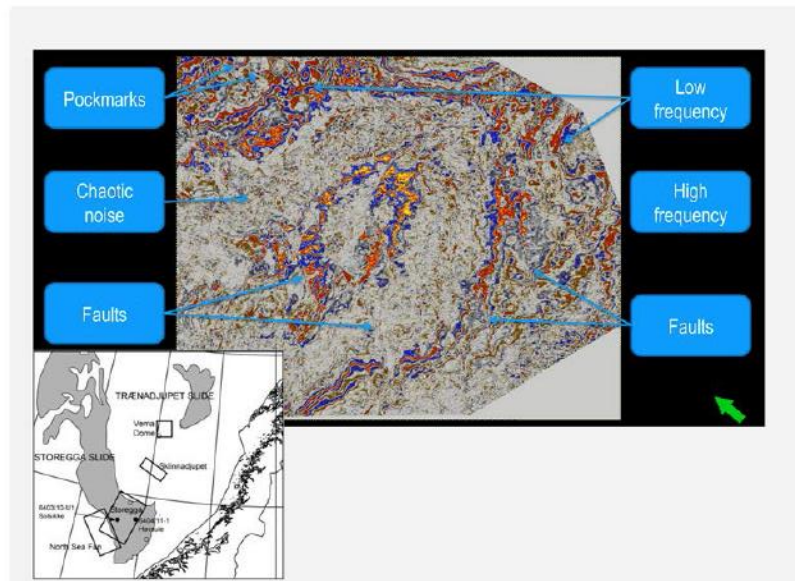


Figure 4: The solsikke dataset used for test our algorithm. A map is shown lower left to highlight the region from offshore Norway where the data is coming from. Also a seismic time-slice is used to highlight the main features that we will be looking at from this data and the challenges that are there to overcome.

This is a heavy faulted dataset with a good representation of the challenges mentioned earlier. It has a clear imaging showing higher and lower frequency changes in seismic data along the time axis. It also has a good representation of the changes in frequency in different traces on the same time, which is due to geological features and their response. As an example, highly faulted areas are more chaotic than non-faulted ones on the same time-slice (See Figure 4). The dataset also has features that are made visible with edge detection attributes, like the pockmarks highlighted in the Figure 4. Another peculiar feature, which is quite common in modern seismic, is the reflectors along the fault plane (See Figure 1). The mentioned features and the fact that it is a highly faulted dataset, make this an ideal case study for the adaptive approach to seismic attributes.

Limitations of the Traditional Approach

In the traditional approach, where one uses a constant operator size, it is often that the calculation results in under or over-sampling. This is due to the changing nature of seismic data. If a small operator is used on low frequency data, then the tendency is to oversample and start detecting noise rather than the edges (fault features) in the seismic image. In the case where a large operator size is used, on high frequency data, then the chances are that features are being skipped or not properly delineated.

The aim of seismic attributes is to perform a certain measurement on seismic, and the use of a fixed operator size causes the measurements to be less accurate. We see this clearly in Figure 5, where we see that highly faulted areas are diluted by oversampling with a too small operator size. The edges of the pockmarks are not picked up properly, and the faults have limited continuity in the low frequency content areas. Overall, although the attribute highlights many fault features it still leaves much to the interpreter, and poses a difficulty for any automated methods as well as uncertainty is added to the measurements.

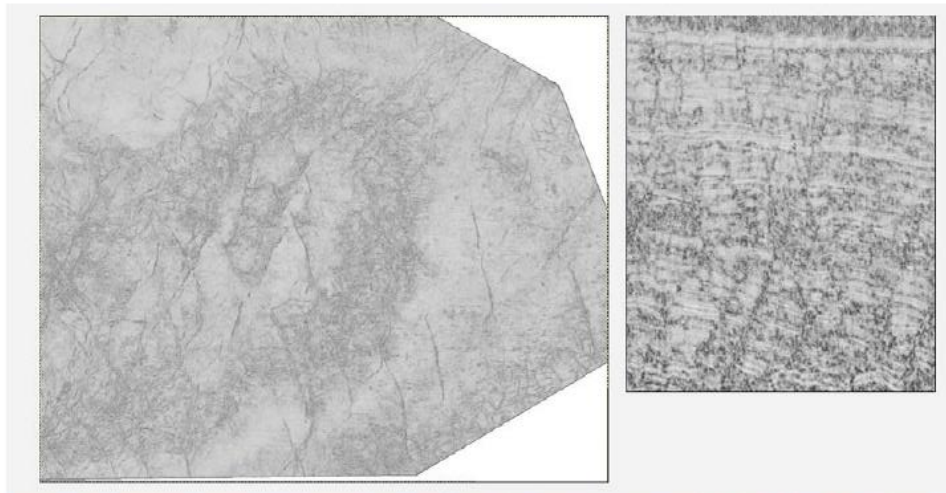


Figure 5: the results of running the traditional edge detection filter using a static operator size. The attribute used here is Amplitude Contrast, which focuses on detecting amplitude discontinuity. The results clearly show that the over and under sampling when trying to pick out the edge features happens not only on a vertical seismic intersection, but also on the time slice. Here we see that features such as the pock marks are not picked up properly due to undersampling, and chaotic regions are oversampled and large amounts of noise are detected as edges.

Enhanced Adaptive Fault Detection Results

As the operator size is adapting to the nature of the seismic, we see a clear enhancement in the edge detection results and a more accurate measurement on the seismic data (see Figure 6). In comparison to the traditional approach, the smaller details are all captured in the high frequency segments. Also the reduction of noise is substantial in the oversampled segments of the data. Overall the results look sharper, more continuous, less noise is detected and with greater detail.

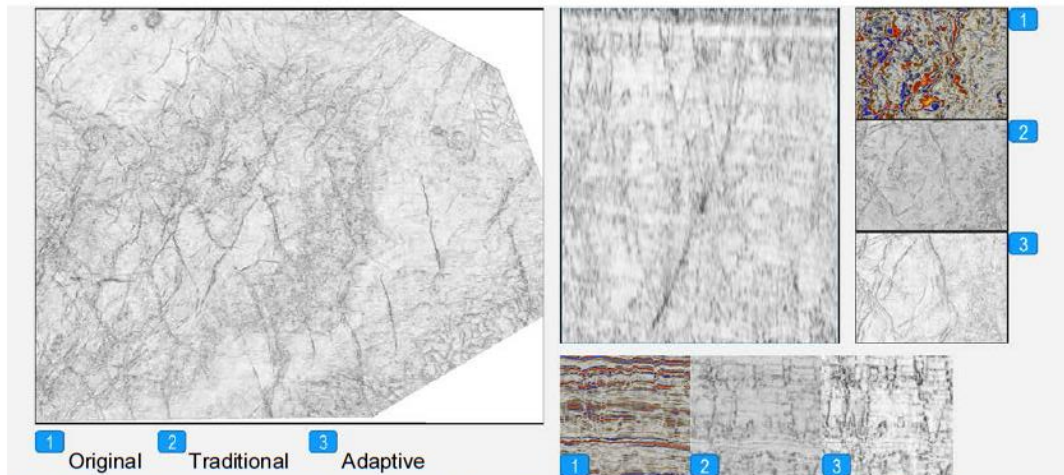


Figure 6: images illustrating the adaptive edge detection Sobel filter on the same seismic intersections as in Figure 5. Here we see a significant reduction in noise detection, and an increase in the detection of amplitude discontinuities and fault features. When looking at direct comparisons, one can see a major significance in the detection, resulting in an overall better measurement of amplitude discontinuity on seismic, and hence a better attribute.

Conclusion

The adaptive approach to edge detection proves to be a more accurate measure than the traditional fixed operator size approach. Overall the results are sharper, more continuous, and less noise is detected. When comparing the two, one can see a major significance in the detection quality, resulting in an overall better measurement of amplitude discontinuity on seismic, and hence a better attribute. This approach is a significant improvement and shows that following the nature of the data filtered gives an advantage.

References

- Aarre V., 2010. Globally Consistent Dip Estimation: SEG Annual Meeting.
- Aqrabi, A. A., and T. H. Boe, 2011, Improved fault segmentation using a dip guided and modified 3D Sobel filter. 81st Annual International Meeting, SEG, Expanded Abstracts 30, 999.
- Aqrabi, A. A., W. Weinzierl, R. Daber, and T. H. Boe, 2012, Directional guided seismic attributes and their use in assisting structural, stratigraphic and lithological interpretation.
- Aqrabi, A.A., Boe, T.H., 2012, Detecting structural geology using dip and directional dip, GEO 2012, Short Abstract
- Bryn et al., 2005. Explaining the Storegga Slide: *Marine and Petroleum Geology*, 22, 11–19
- Chopra, S., and K. J. Marfurt, 2005, Seismic attributes—A historical perspective, *Geophysics*, 70(5), 3SO.
- Daber, R. E., and Boe, T.H., 2010, Using azimuthal curvature as a method for reducing noise in poststack curvature volumes: 72nd Conference and Exhibition, EAGE, D024.
- Iske and Randen, 2005. *Mathematical methods and modeling in hydrocarbon exploration and production*: Springer-Verlag
- Nissen, S. E., T. R. Carr, K. J. Marfurt, and E. C. Sullivan, 2009, Using 3-D seismic volumetric curvature attributes to identify fracture trends in a depleted Mississippian carbonate reservoir: implications for assessing candidates for CO₂ sequestration, *AAPG Studies in Geology*, 59, 297–319.
- Pedersen, S. I., Randen, T., Sonneland, L., and Steen, O., 2002, Automatic fault extraction using artificial ants: 72nd Annual International Meeting, SEG, Expanded Abstracts, 21, 512–515.
- Rijks, E. J. H., and J. C. E. M. Jauffred, 1991, Seismic interpretation 29; attribute extraction; an important application in any detailed 3-D interpretation study, *The Leading Edge*, 10, 11–19.

BIBLIOGRAPHY

- [1] Schlumberger, "Software," [Online]. Available: http://www.software.slb.com/store/_layouts/SLB/Pages/ProductDetailPage.aspx?pid=AASI-B1. [Accessed 10 February 2015].
- [2] A. A. Aqrawi and D. S. Barka, "Adaptive Sobel Based Edge Detection for Enhanced Fault Segmentation," in *IPTC*, Doha, 2014.
- [3] A. A. Aqrawi and D. S. Barka, "Enhanced Fault Segmentation using an Adaptive 3-D Sobel Filter," in *AAPG*, Pittsburgh, 2013.
- [4] A. A. Aqrawi, D. S. Barka and W. Weinzierl, "A Hybrid and Adaptive Attribute for Noise Reduction in Post-Stack Seismic Data," in *75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013*, 2013.
- [5] B. Biondi, *Three Dimensional Seismic Imaging*, Tulsa, Okla: Society of Exploration Geophysicists, 2006.
- [6] "Seismic Imaging," Delft University of Technology, 2012.
- [7] S. Chopra and K. J. Marfurt, "Seismic Attributes a Historical Perspective," *Geophysicists*, vol. 70, no. 5, pp. 3SO-28SO, 2005.
- [8] T. Randen and L. Sonneland, "Automated Geometry Extraction form 3D Seismic Data," in *SEG Annual Maeeting*, Dallas, 2003.
- [9] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, Newnes, 2002.
- [10] I. Sobel, *An Isotropic 3x3 Image Gradient Operator*, Stanford, 2014.
- [11] A. A. Aqrawi and T. H. Bø, "Improved Fault Segmentation Using a Dip Guided and Modified 3D Sobel Filter," in *SEG*, San Antonio, 2011.
- [12] Schlumberger Information Solutions, *Petrel Introduction G&G*, Schlumberger, 2010.

- [13] Schlumberger Information Solutions, *Petrel Geophysical Software*, 2014.
- [14] Schlumberger Information Solutions, *Petrel Seismic Interpretation*, 2014.
- [15] MathWorks Inc, "MathWorks.com," [Online]. Available: <http://se.mathworks.com/products/matlab/>. [Accessed 7 February 2015].
- [16] CREWES , "crewes.org," [Online]. Available: <http://www.crewes.org/>. [Accessed 6 February 2015].
- [17] Society of Exploration Geophysicists, "SEG," [Online]. Available: <http://www.seg.org/seg>. [Accessed 14 April 2015].
- [18] dGB Earth Sciences B.V, "Open Seismic," [Online]. Available: <https://opendtect.org/osr/pmwiki.php/Main/NetherlandsOffshoreF3BlockComplete4GB>. [Accessed 2 March 2015].
- [19] Mathworks, "Documentation," [Online]. Available: <http://se.mathworks.com/help/matlab/ref/fft.html>. [Accessed 10 4 2015].
- [20] G. Strang, "This is the most important numerical algorithm of our lifetime...", *American Scientist*, vol. 82, no. 3, p. 253, 1994.
- [21] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proceedings of IEEE*, vol. 66, no. 1, pp. 51-83, 1978.
- [22] D. Hermansen, F. S. Imaging, T. Elboth and F. Geoteam, "Attenuation of Noise in Marine Seismic Data," in *SEG Annual Meeting*, Houston, 2009.
- [23] M. Basu, "Gaussian-based edge-detection methods-a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 32, no. 3, pp. 252-260, 2002.
- [24] D. A. Forsyth and J. Ponce, *Computer Vision a Modern Approach*, Pearson, 2012.

- [25] Y. Lu and R. C. Jain, "Behavior of Edges in Scale Space," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 4, pp. 337-356, 1989.
- [26] G. R. Arce, *Nonlinear Signal Processing: A Statistical Approach*, New Jersey: Wiley, 2005.
- [27] K. Haris, S. N. Efstratiadis, N. Maglaveras and A. K. Katsaggelos, "Hybrid Image Segmentation using Watersheds and Fast Region Merging," *Image Processing, IEEE Transactions on*, vol. 7, no. 12, pp. 1684-1699, 1998.
- [28] A. A. Aqrabi and T. H. Boe, "Imaging structural geology with dip and directional dip," in *Geo*, Bahrain, 2012.
- [29] A. Fabijanska, "Variance Filter for EWdge Detection and Edge-Based Image Segmentation," in *IEEE Xplore*.