University of Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: Master of Science in Computer Science	Spring semester, 2012 Open access
Writer: Nour Alabbasi	…… (Writer's signature)
Faculty supervisor: Tom Ryen External supervisor(s): Ahmed Adnan Aqrawi	
Title of thesis: Seeded Growing Algorithms for Salt Body Segmentation in Post-Stack Seismic Data	
Credits (ECTS):	
Key words: Seeded growing, salt bodies, segmentation, post-stack seismic data, Sobel filter, smoothing filters, automatic detection, edge detection , discontinuous boundary, hybrid smoothing.	Pages: ……89…… + enclosure: …118 + CD…… Stavanger, ……13.06.2012…….. Date/year

Seeded Growing Algorithms for Salt Body Segmentation in Post-Stack Seismic Data

Nour Alabbasi

June 13, 2012

Abstract

For a more accurate geological model, it is crucial to have proper segmentation and delineation of structures. Salt structures are known to be difficult to segment given their chaotic nature; however, there are methods that isolate the salt borders well. Delineation of salt structures in post stack seismic data is an inherently difficult problem and is of great value when detecting potential reservoirs. This is the case in regions such as the Gulf of Mexico. There exist several approaches where an image filtering algorithm is used to detect such structures; such is the case with well known seismic attributes like variance, coherence, amplitude contrast, etc. These attributes usually are a good indication of the flanks of the salt structure but struggle in segmenting the salt body as a whole.

In our work, we propose an automated seeded growing method that will segment the salt bodies from the other structures in seismic data. We Used the edge volumes as input to get a good indication of the flanks and borders of our salt structures. We build our method upon the conventional seeded growing approach, and expand it with hybrid smoothing methods such as median, mean, Gaussian, adaptive median. Our alternating criteria, in this case, is the amount of chaos detected. Finally we detected the discontinuous boundary by evaluating the growing directions of our seed points and terminate based on unbalanced behavior.

We implemented our method using Matlab and tested it using dataset from the Gulf of Mexico. By running our algorithm, we got an automated detection of the seeding within the salt bodies, clear segmentation of the salt bodies and a termination at the flanks. The results show that even given the noisy nature of the salt images, the method is able to segment the salt bodies entirely and consistently. We managed to terminate only at the boundaries of the salt bodies and not before or after. This gives a more consistent segmentation, which is evident even in the case where the structures boundaries are disconnected. Our approach of estimating the boundaries when not present and looking at unbalanced growing, results in a consistent segmentation nonetheless. Evaluating all the stages in our algorithm, we find that the computation complexity is bounded by $O(NlogN)$

The approach of a seeded growing algorithm to segment salt bodies proves to be useful, and given a proper input volume is able to isolate the salt body as a whole. The approach presented here, with the modifications, is effective at this task only when combined with methods of noise removal/reduction and boundary termination estimation for existing and discontinuous boundaries. The hybrid smoothing approach has also proven to be a very useful combination with the growing method as it alternates smoothing techniques to optimize the segmentation.

Acknowledgement

This thesis, is the result of a Master thesis project assigned by the Department of Electrical and Computer Engineering at the University of Stavanger. I would like to thank my supervisor Tom Ryen for invaluable support and feedback throughout the entire thesis.

I would like to thank my supervisor from Schlumberger Ahmed Adnan Aqrawi. He has been an inspiration with his great understanding and dedication to the entire thesis. Given his generosity and encouragement all the resources needed for this project where made available. Also the support in providing me with new ideas, example source code and a set of seismic data.

A special thanks to my parents and my sister Carmel for their help and support.

Nour Alabbasi
Stavanger

CONTENTS

LIST OF FIGURES

LIST OF TABLES

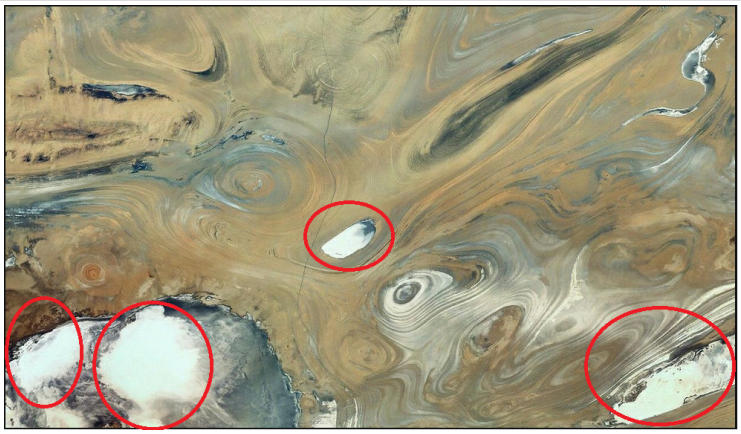
LIST OF ALGORITHMS

Chapter 1

INTRODUCTION

The oil and gas industry nowadays are focusing more on exploring unconventional and more complex areas. To do so, a lot of research is being done on seismic imaging to produce better visuals of the sub-surface. Having more detailed seismic data opens up for greater possibilities in image enhancement and processing techniques, which will better aid a geoscientist in locating potential reservoirs. Such filters used in the industry are deemed seismic attributes.

Figure 1.1 Satellite image of salt tectonic features, Great Kavir Basin, Semnan Province, Iran [19]



One structure with particular interest is the salt body (See Figure 1.1), which is not only hard to image in the sub surface, but also difficult to detect and interpret correctly. This is mainly due to its noisy nature and ambiguous borders and shapes. However, salt structures are of great value when detecting potential oil and gas reservoirs. There exist several approaches where an edge detecting

seismic attributes are used to detect such structures; such is the case with well known attributes in the coherence family (Chopra and Marfurt, 2007)[10]. These attributes usually are a good indication of the salt body borders, but struggle in segmenting the salt body as a whole.

Seed growing is a known method within image processing and has several applications such as medical, face recognition, and even in seismic for surface detection (Adams and Bischof, 1994) [1]. In this work, we intend to use the seeded growing approach in segmentation of salt bodies in seismic data to better aid interpreters in understanding these salt structures. We have expanded the traditional seed growing approach with the use of other filtering methods as the growing takes place to target the delineation of salt bodies. We have relied on smoothing and noise reduction filters such as the median and the mean (Lee and Kassam, 1985)[25] (Chan et.al, 2005)[9] (Linville and Meek, 1995)[26] as well and edge/boundary detection methods like amplitude contrast (Aqrawi et.al., 2011) [3] [5]. Another aspect that is of interest when detecting these salt bodies is the discontinuous nature of their boundaries and how to terminate to avoid growing into other structures (Cohen and Coifman, 2002)[12]. We have also built further upon this by adding some intelligence to the algorithm where is will decide when to use certain method to optimize the growing accuracy.

1.1 Goals

In this work, we want to segment the salt bodies in post stack seismic data. To aid geoscientist in their interpretation of the salt bodies, we suggest a method that separate the salt body from the flanks. To achieve this, we have three main goals that should be addressed:

1. To develop algorithm using the seeded growing method to segment the salt bodies in post stack seismic data.
2. To implement smoothing filters to remove the noise before detecting the salt bodies borders. This is to ensure that the segmented body will be solid and not porous.
3. To estimate the missing borders of the salt bodies and define termination conditions to guarantee that the detected body is consistent in comparing to our knowledge to the salt structure.

1.2 Our Contributions

The contributions of this thesis include:

- Patented Seeded growing algorithm for salt body segmentation in post stack seismic data, by using the combination of image processing tech-

niques such as smoothing, edge detection and boundary estimation. (See Appendix B for more details regarding patent IS122633USPSP_20120529).

- Presenting different approaches to automate the seed picking such that user does not need to manually choose the seeds.
- Implementation of the edge enhancement techniques similar to edge evidence [2], but focuses on circular shapes rather than line segments.
- Implementation of several noise removal filters and comparison study on their effect on the seismic data.
- Implementing a hybrid approach for the noise removal filters and switching between them according to the neighborhood filtered within our algorithms.
- Presenting the directional history approach for the seeding points to determine termination along a non-existing border for non-complete boundaries.

1.3 Thesis Outline

This thesis will be structured as follows:

Chapter 2: Relevant background material is highlighted and explained allowing the reader to understand the work done in this thesis. Topics such as: seismic imaging, previous work and the Petrel software.

Chapter 3: Relevant background of the algorithms that we will build our work on. This includes the seeded growing algorithm and different smoothing filters such as: median , adaptive median, mean and Gaussian filters.

Chapter 4: Describes how the implementations in this thesis are performed, and why certain implementation decisions were made. Here one will also find the different steps that we went through to develop the final algorithm.

Chapter 5: In this Chapter, Results when running the different algorithms are presented and discussed. The main focus here is on comparing the implementations and the different results when adding more conditions. In addition to an analysis of the algorithms performance.

Chapter 6: Here the conclusion of the work performed are presented. There will also be suggested future work within the field.

Glossary: contains explanations of geophysics concepts relevant and were included within our work.

Appendix A: an extended abstract (short paper), written to be sent to SEG annual meeting in Texas 2013 confernce in USA. The actual paper will be written later as the deadline is in September.

Appendix B: This is a provisional patent filed to protect our work in the next year. It was filed in 29.05.2012 with a patent number: IS122633USPSP_20120529.

Appendix C: this is the poster that was presented at UiS in 30.05.2012

We have explained in this report the implementation issues that we did in our work. The full code for all the mentioned algorithms, in addition to the synthetic data and seismic data can be found in a CD attached to this report. The seismic data was given by Schlumburger , so the access is restricted and in case of any further use to this data, a permission from Schlumburger is needed.

Chapter 2

GENERAL BACKGROUND

In this Chapter, we will introduce background material in the topics that are relevant to our work. The reader can skip the sections that are familiar. The chapter will be divided into four sections.

In section 2.1, we will present the Seismic Imaging and the connection to Image Processing. In addition to an explanation of the Salt Structures in seismic images, the importance of salt in the oil industry, and the main characteristics of salt that make it an interesting problem.

In section 2.2, we will show previous contributions related to our work. The first one, is about seismic data interpretation using the Hough transform and principle component analysis. In this work, one of the goals is similar to ours, which is to identify shapes associated with complex salt bodies in seismic profiles. The methodology was different from ours. However, it was useful to compare and check the different methodologies focusing on the same goal. The second, is about edge detection analysis using 3D seismic data. In this work, the salt segmentation was not mentioned directly but used edge detection methods can be applied to our task. Using the derivatives for detection of structural and stratigraphic discontinuities has similarities to our approach.

In section 2.3, we will introduce the Petrel Software as an integrated workflow tool that includes most of the reservoir modeling procedures and allows specialized geoscientists to cooperate seamlessly. Our input data is going to be preprocessed using Petrel. Furthermore, We introduce the Ocean for Petrel API as an object oriented framework which is built on .NET and which uses C as its language. This is because in our future work, we hope to integrate the implementation of our algorithms in the Petrel Workflow using Ocean.

In the last section 2.4, we will have a short description to Matlab, image processing toolkit and our motivation to implement our algorithms there.

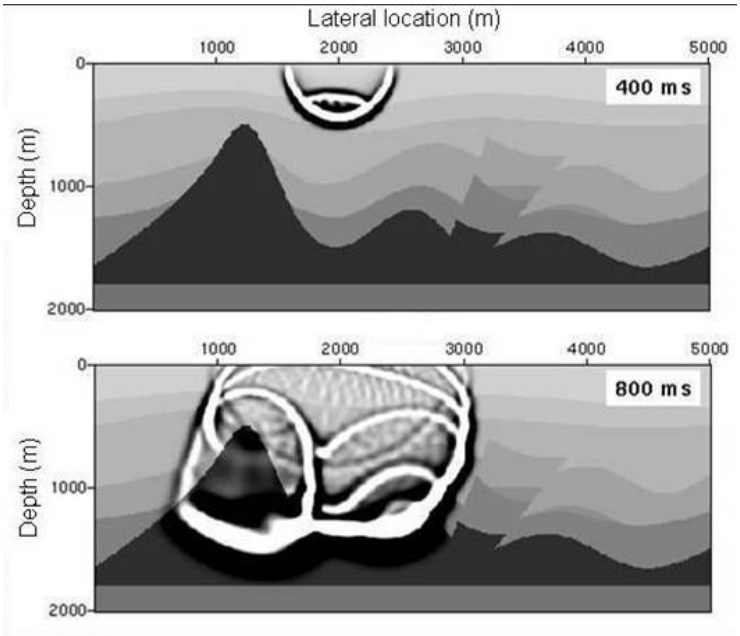
2.1 Seismic imaging and salt tectonics

In this section, we will briefly explain the definition of seismic imaging and the connection to Image Processing. Then we will focus more on our motivation in this work by explaining the importance of salt structures in seismic imaging and the different tectonic structures that make it a complex and interesting area to work with.

2.1.1 Seismic imaging

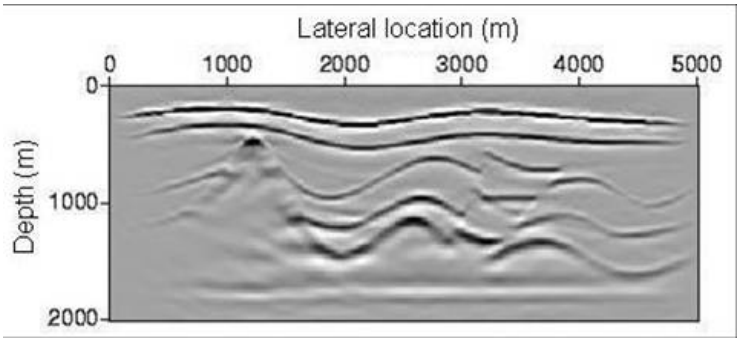
Seismic images can reveal significant pictures of the earth's subsurface geology [8]. These pictures can be used to understand and discover structural features at deep depths, and as a result help us to identify the location of oil and gas deposits. Seismic imaging straights a forceful sound source into the ground to estimate subsurface conditions and to possibly sense high concentrations of contamination. Receivers called geophones, analogous to microphones, pick up "echoes" that reflect back up through the ground and record the intensity and time of the "echo" on computers (See Figures 2.1 and 2.2)

Figure 2.1 Propagating sound source [31]



Data processing turns these signals into images of the geologic structure. During the survey process, the reflections provide a three-dimensional digital model of

Figure 2.2 Processed data from acquisition from Figure 2.1 [31]



the subsurface. This information can be used to identify preferential flow paths, determine the placement and screening of wells, and help select a remediation technology.

Seismic images frequently show patterns with a layered structure due to the depositional nature of the subsurface (See Figure2.3). A seismic amplitude section with seven texture elements (texel) highlighted to show their textural differences [10]. A. Isolated amplitude anomaly; B. Moderate-low amplitude and low continuity; C. Chaotic, hummocky, moderate amplitude, and low continuity; D. Low amplitude and low-moderate continuity; E. Low-moderate amplitude and moderate continuity; F. High amplitude and high continuity; G. Low-amplitude, low continuity, and massive. Although these features can be recognizable subjectively by visual inspection on this specific line, they may not be as easily recognizable in other lines and it is difficult to isolate and map them quantitatively in 3D by visual inspection.

In image processing, a pattern with a certain regularity or structure is called a texture. The description of the 'layered' textures in seismic images can be split up in two parts [6]. One part is the geometrical description of the structure, the other part is the description of the signal perpendicular to the layered structure. Examples of geometrical properties are the orientation and the curvature of the layered structure. An example of a property of the perpendicular signal is its characteristic frequency. In the case of a seismic image, the perpendicular signal is determined by the change in the acoustic impedance of the subsurface rock, convolved with the seismic wavelet. This convolved signal is usually described by using a time-frequency representation [39], [37].

The individual layers of a layered texture in an image (See Figure 2.4) can locally be approximated with isophotes. Isophotes are curves and surfaces with a constant intensity value. For the geometrical description of these isophotes, we will focus on the analysis of the image intensities directly in our algorithm. The relevant properties of the patterns that we will detect, will depend on the scale of the analysis. At the smallest scale, the level of the individual points of

Figure 2.3 Seismic Texture [10]

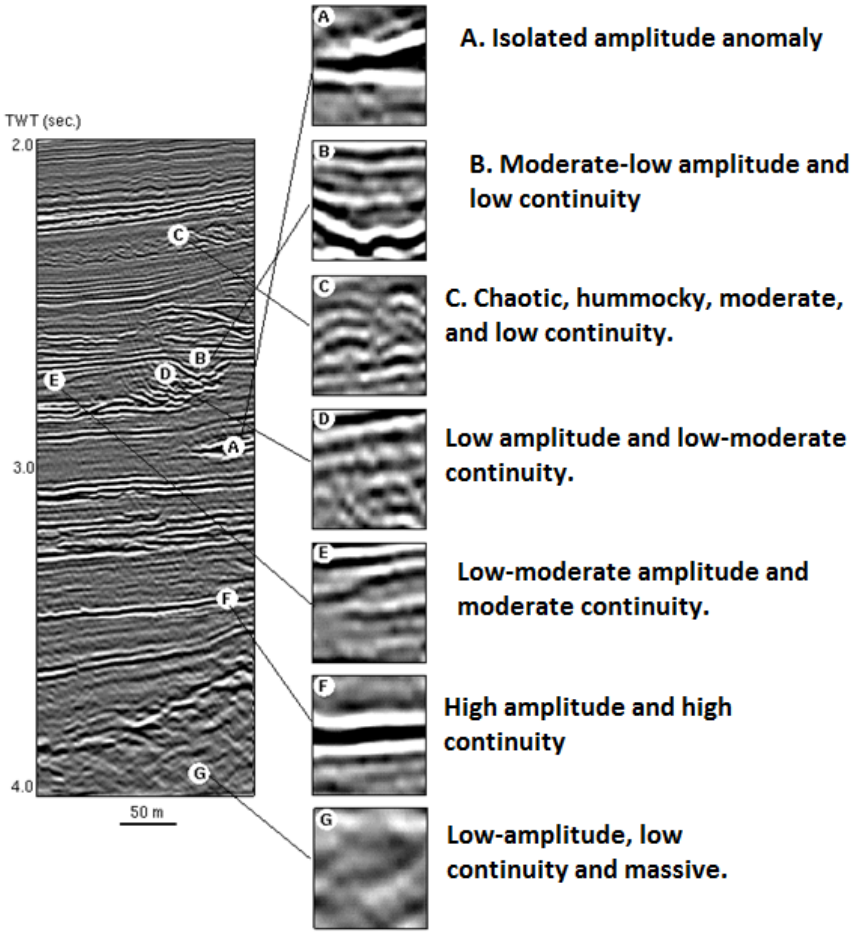
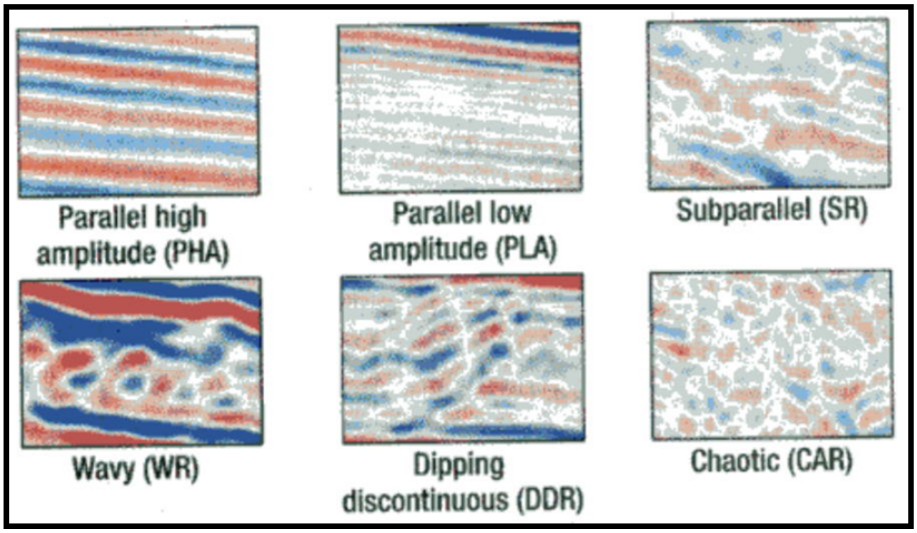


Figure 2.4 Seismic texture [22]



the image, only the intensity of the image point itself can be measured. The description of the isophotes in general becomes very complex. In general, the optimal scale for the estimation of some feature may vary over the image. The algorithms may also be present at several scales simultaneously, by defining different operators in the filters that we will use in our work. In all cases, our work requires the estimation at multiple scales.

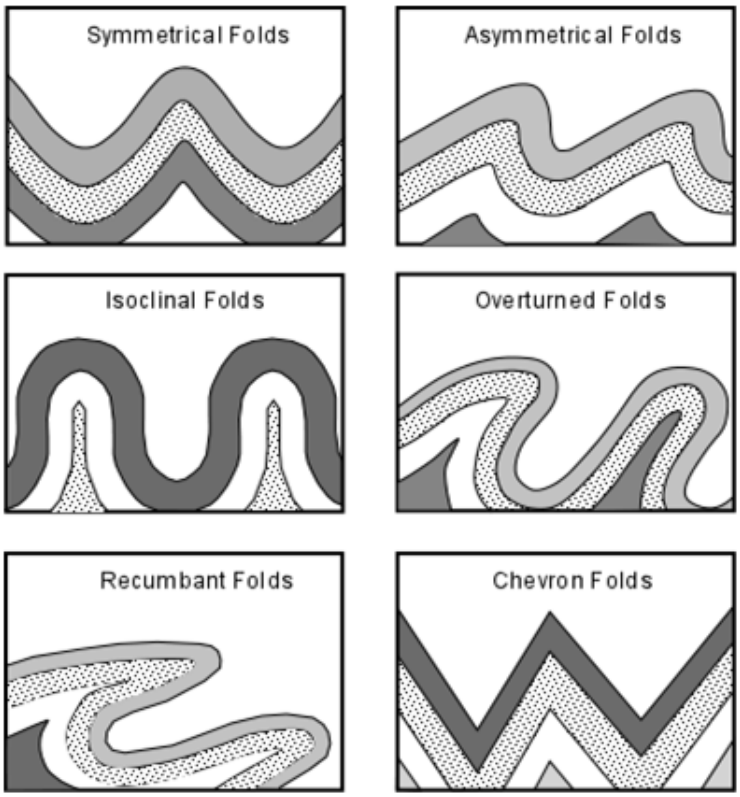
2.1.2 Salt tectonics

One of the most difficult zones to image is the area under the edges of salt sheets. Discontinuities in both structure and amplitude are often observed in the data beneath the transition zone. Even with the vast improvement in seismic imaging with the use of 3-D prestack depth migration, there are problematic areas were interpretation of the subsalt structure is unclear [13]. In spite of all that, a lot of research is now considering this area in both geophysics and image processing arenas, because a significant proportion of the world's hydrocarbon reserves are found in structures related to salt tectonics, including many in the Middle East, the South Atlantic passive margins (Brazil, Gabon and Angola) and the Gulf of Mexico.

Salt tectonics focus on the geometries and processes associated with the presence of significant thicknesses of evaporates containing rock salt within a stratigraphic sequence of rocks. This is because both the low density of salt, which does not increase with burial, and its low strength [46].

The salt can be structured in three main shapes [29]:active, passive and reactive. The active structure will increase the possibility of salt structures developing. If the tectonics were extensional (can be called thrust tectonics), it will reduce the strength of the overburden and thin it. The bulking of the overburden layer will allow it to rise into cores of anticlines which will form the salt domes. The whole process will be effected by forces of tectonic movements, which causes salt bodies to be pushed through its overburden as forceful diaprirism. This folding can take different shapes (See Figure 2.5).

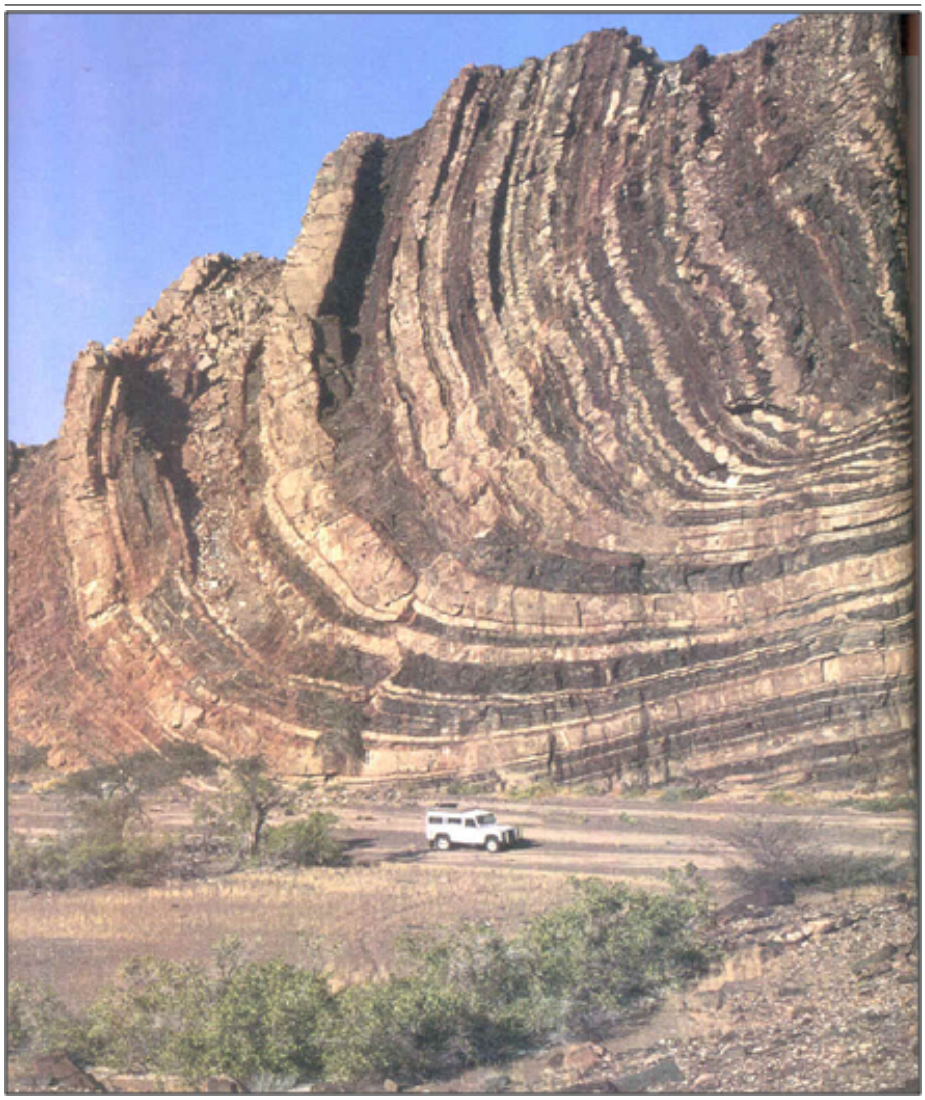
Figure 2.5 Types of folding [43]



Many salt diapirs may contain elements of both active and passive salt movement. This will vary depending on the location nature (the pressure difference) see Figure 2.6 from Namibia to create different characters in the different geographical areas. The middle east nature will definitely be the same as the Gulf of Mexico for instance.

The passive structures may form during continued sedimentary loading, without any external tectonic influence, as a result of gravitational instability. When the

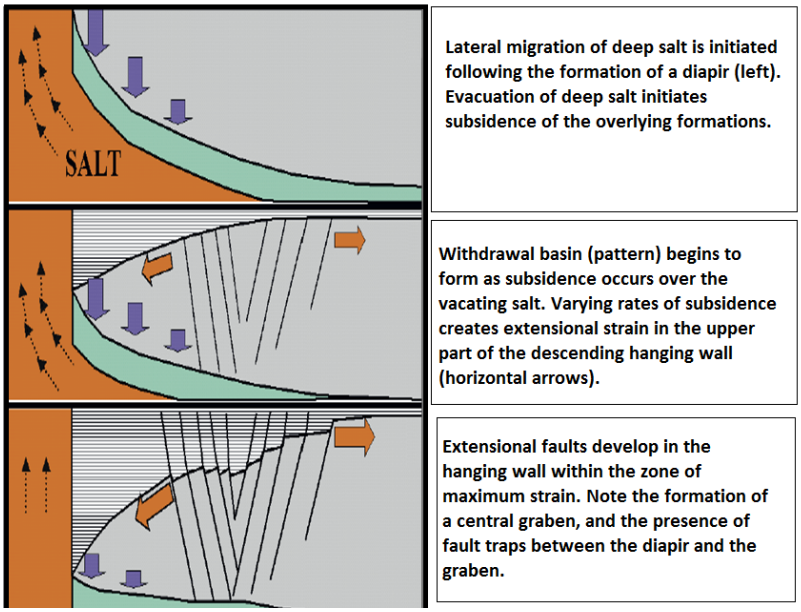
Figure 2.6 Lower Ugab Valley In Namibia [41]



overlying layers become denser, the weak salt layer will tend to distort into a characteristic series of ridges and depressions and the salt will continue to move away from them into the ridges. Later on, the diapirs lean to initiate at the junctions between the ridges, this will be differentiated along the ridge system continuously until the salt supply is drained. During this progression , the top of the salt diapir remains at or near the surface (downbuilding). One of the best examples of a purely passive salt structure is located in Germany.

The reactive salt structure happens when the salt layers do not have the conditions to develop the passive structure, so the salt can still move into low pressure areas around developing folds and faults (it is really important in the interpretation cases to classify the faults and the salt in the same sample).

Figure 2.7 Geological model [23]



The geological model in Figure 2.7 illustrates lateral migration of deep salt is initiated following the formation of a diapir (The Top to the Left). Evacuation of deep salt initiates subsidence of the overlying formations. Withdrawal basin (pattern) begins to form as subsidence occurs over the vacating salt. Varying rates of subsidence creates extensional strain in the upper part of the descending hanging wall (horizontal arrows). Extensional faults develop in the hanging wall within the zone of maximum strain. Note the formation of a central graben, and the presence of fault traps between the diapir and the graben.

Taking a time slice (see Figure 2.8), we can note that the intense ring faulting associated with salt withdrawal in the right and the left. By the one in the middle, there are many ring faults.

Figure 2.8 Time slice through coherence volume (East Texas) (courtesy of WesternGeco)

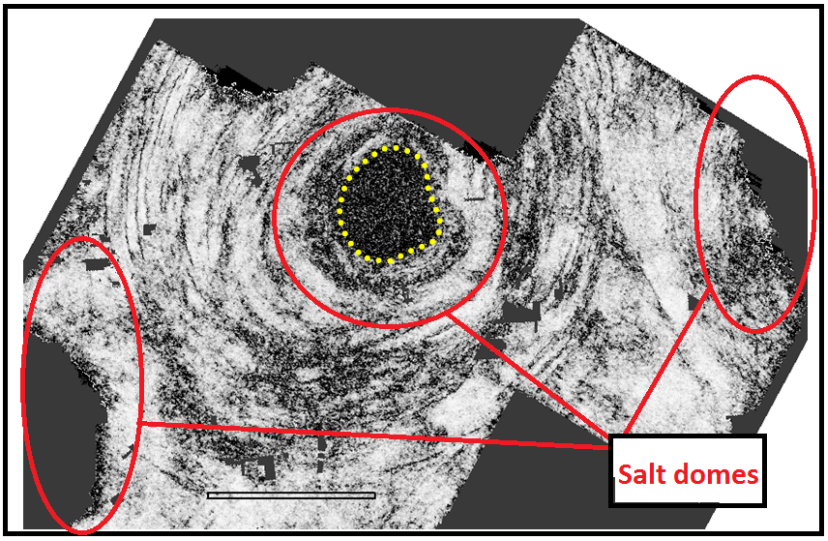
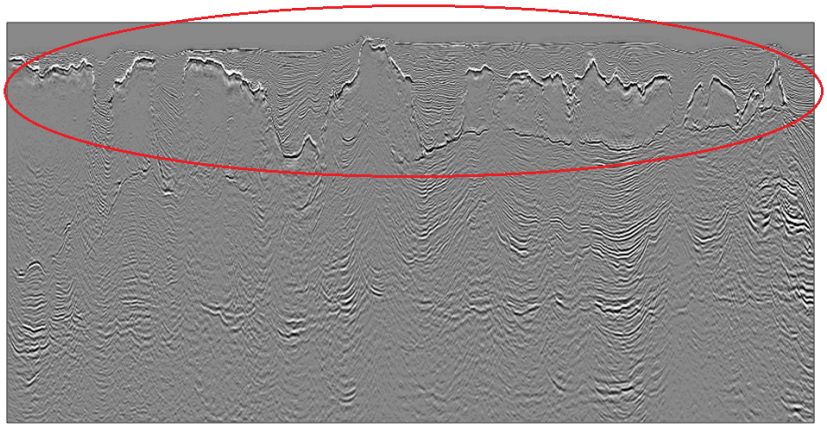


Figure 2.9 Salt Seismic response (courtesy of WesternGeco)



Figures 2.9, 2.10 show the seismic response for the salt. In Figure 2.9, the seismic response from the side, we can see that the lines are not continuous. Looking to the seismic from the top in Figure 2.10 shows how the salt domes look like circle in the middle with the change in the color due to the intensity difference.

Figure 2.10 Salt Seismic response (courtesy of WesternGeco)

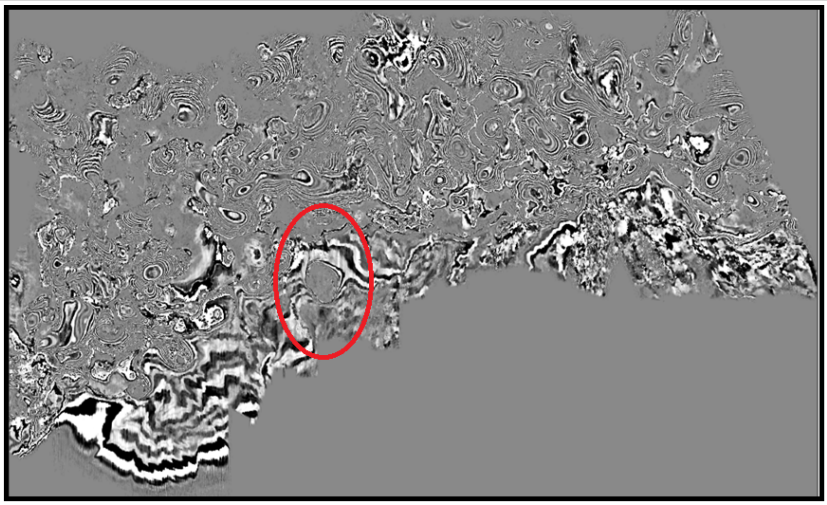


Figure 2.11, shows the data seismic from the top after filtering. Variance showing faulting around a salt dome in map view. Again in this shape we can see that the salt edges in seismic data will have some disconnected areas that can effect the detection method. And this should be taken in consideration in our work.

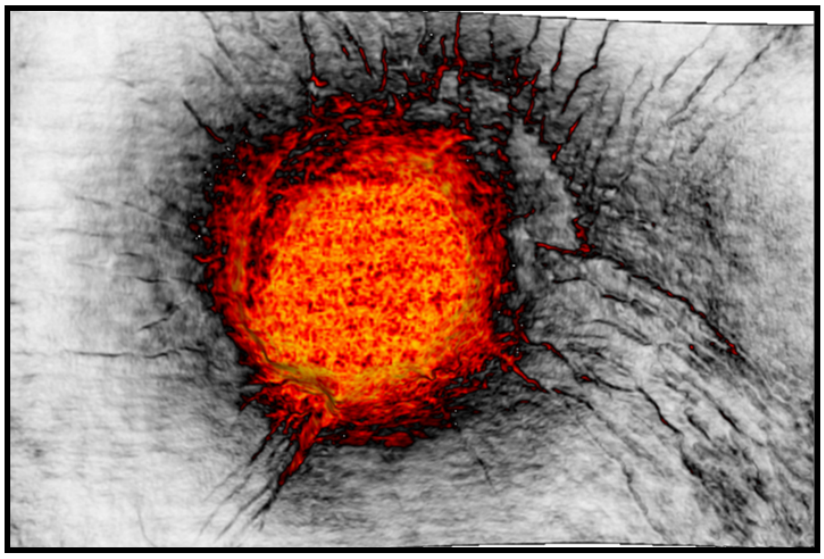
2.2 Previous Work

It is well known to the geophysical community that the nature of the subsalt (lack of the resolution and poor structure identification) in seismic data can cause rigorous technical and economical problems. Under such circumstances, seismic interpretation based only on the human-eye is inaccurate. In addition to the need for long experience and full understanding of the global variables that can be diverse resting on the different geographical characteristics for the shooting region.

Furthermore, petroleum field development decisions and production planning depend on good-quality seismic images that generally are not feasible in salt tectonic areas.

In this section, we will present two previous works, that are related to our task.

Figure 2.11 Salt dome showing radial faulting (courtesy of WesternGeco)



The first work is focusing on the seismic data interpretation using the Hough transform and principle component analysis to detect the salt bodies (specifically the parabolic shape). The second, is Edge detection and stratigraphic analysis using 3D seismic data, with different methods to detect the different patterns in the seismic data.

2.2.1 The seismic data interpretation using the Hough transform and principle component analysis(Orozco 2010)[14]

The main theoretical background in this work was depending on the following: morphological erosion, region growing and especially a generalization of the Hough transform (closely related to the Radon transform) are applied to build parabolic shapes that are useful in the idealization and recognition of salt domes from 2D seismic profiles.

In a similar way, principal component analysis (PCA) is used to identify shapes associated with complex salt bodies in seismic profiles extracted from 3D seismic data.

The previous methodology was focusing mainly to detect the parabolic shapes of the salt. The biggest disadvantage would be that the parameter space resulting from the application of the generalized Hough transform would be represented by a higher dimensional space. In addition to the fact that the salt is taking different shapes in the different places , so this method can be applied to specific

setting of data, but can be improved and extended to include the rest of the shapes. Yet, some kind of salt will not take a regular geometry shape where we can pre-calculate or expect the behavior.

A PCA system in the same method working directly with 3D data should improve the accuracy of the detection process itself, although it would also increase significantly the time required for the training of the system and the detection process itself.

2.2.2 Edge detection and stratigraphic analysis using 3D seismic data (Y. Luo 1996)[45]

In this paper, two methods for detecting stratigraphic boundaries in 3D seismic data are described. The difference method is based on calculating the local trace to trace difference of the seismic signal. The derivative method calculates the spatial derivative of instantaneous phase to locate points of rapid phase change. These edge detection techniques provide a variety of new attributes that can be generated early in the interpretation work process to aid interpretation.

Edge detection technology can be employed early in the interpretation work process to highlight faults and stratigraphic boundaries within the 3D seismic data volume. There are a number of advantages to using edge detection technology as an integrated part of the work flow. Structural and stratigraphic discontinuities can be identified without any prior interpretation, which results in reduced interpretation bias and reduced interpretation cycle time. Edge detection technology also enables rapid and high resolution evaluation of 3D seismic data at all stages of exploration and exploitation of reservoirs, for example the rapid appraisal of reconnaissance 3D seismic data. A number of the edge detection attributes can be used for better risk assessment of structural traps and better delineation of reservoir barriers for depletion planning.

In this work, nothing specific toward salt detection was mentioned. Yet the concept of the edge detection methods can still be applied for salt body segmentation.

2.3 Petrel

Petrel is an integrated workflow tool that includes most of the reservoir modeling procedures and allows specialized geoscientists to collaborate seamlessly (See Figure 2.12).

In this section, we will present shortly the Petrel Software. The focus will be mainly on the Petrel geophysics component and the Petrel exploration geophysics tools. At the end of this section, longer discussion for the Petrel Seismic interpretation workflow will be presented. Our goal is to present for the readers

who are not familiar with Petrel, the main functions provided in the seismic interpretation workflow, since we can see that our contribution during this report will fit in this workflow. All the information related to this section is provided by Schlumberger Information Solutions [34],[35] and [36] .

2.3.1 Petrel geophysics component

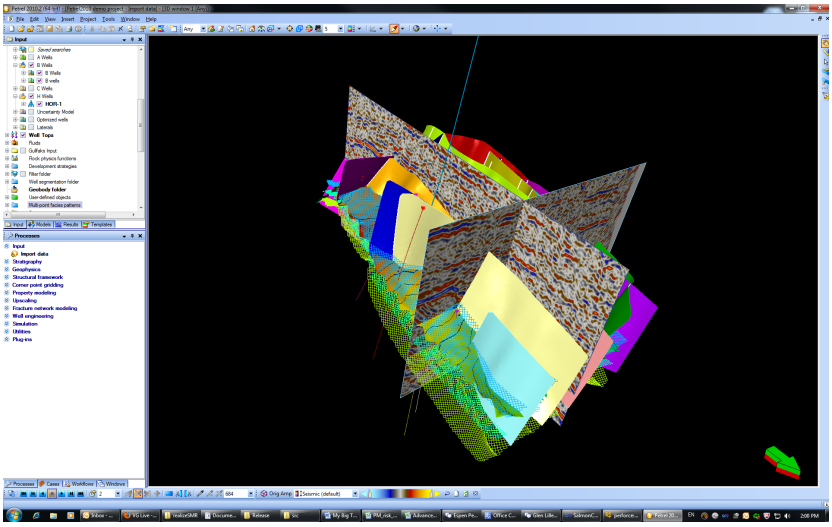
provides a full spectrum of geophysical workflows, including 2D and 3D interpretation, a full set of complex volume and surface attributes including ant tracking for the identification of faults and fractures, volume interpretation (geobody detection) with seismic cross plotting and classification, domain conversion, and the modeling-while-interpreting functionality, which enables interpreters to build a structural framework while doing their interpretation. Petrel geophysics is a solution that supports seismic data from different coordinate systems, from regional exploration to reservoir development.

2.3.2 Petrel exploration geophysics

Petrel exploration geophysics provides the following:

- Petrel seismic interpretation: visualize and interpret regional 2D and 3D seismic data manually or use advanced auto-tracking techniques. Interactively create attribute maps of horizons or intervals (See Figure 2.12).
- Petrel seismic volume rendering & geobody extraction: interactively blend multiple seismic volumes, isolate areas of interest, and then instantly extract what is visualized into a 3D object called a geobody.
- Petrel seismic attribute analysis: generate and analyze seismic attributes to enhance information that might be subtle in traditional seismic, leading to a better interpretation of the data.
- Petrel domain conversion: quickly perform domain conversion backwards and forwards between time and depth. Create your velocity models directly in Petrel or import from any third party application.
- Petrel seismic sampling: convert your seismic data to depth and resample the seismic attribute into the 3D structural grid as a property. Petrel Synthetic Seismograms.
- Bridge the gap between your time and depth domains.
- Petrel automated structural interpretation:by focusing on structural geology rather than conventional segment picking, Automated Structural Interpretation reduces conventional interpretation time while increasing your level of geological detail, structural awareness and reservoir understanding.

Figure 2.12 Visualization in 3D Petrel



- Petrel automated structural interpretation Ant-Tracking
- Classification & estimation: estimate well logs, surfaces, seismic volumes, and 3D property models using neutral network technology.
- Petrel well path design: design well paths, identify surface locations, pick targets, and adjust trajectories dynamically in a 3D canvas to find the optimal solution.

2.3.3 Petrel seismic interpretation

Petrel seismic interpretation software seamlessly combines the workflows of 2D interpretation with the visual and performance benefits of 3D volume interpretation. You can leverage an interpretation environment unified with geology, reservoir modeling, and reservoir engineering domains and have the ability to rapidly interpret seismic data and compare the results with other data in your project. Effortlessly move from interpretation to structural model building to property modeling and back, eliminating the gaps and inevitable knowledge and data loss of traditional systems that require handoffs from one technical domain to the next.

- 2D seismic interpretation: unlock more information from 2D seismic data by applying techniques that are typically reserved for 3D interpretation, including opacity control in the 3D canvas.
- 3D seismic interpretation: Visualize and interpret massive amounts of 3D data without loading all the data to RAM. Combine the traditional line-

by-line approach with the latest algorithms and tools, including amplitude- and waveform-based tracking, for 3D volume interpretation. Rapidly identify stratigraphic or structural features, and then interpret horizon and fault though the volumes.

- 2D/3D multi-volume interpretation: Interpret across multiple 3D and 2D surveys-either in the interpretation or in 3D windows-to gain the best understanding in the shortest time. Interpret the same event across multiple surveys, and grid, contour, and map whole or partial events for individual surveys.
- Modeling-while-interpreting (MWI) functionality: Pick faults in the background while Petrel software grids faults and applies fault connection rules, producing an accurate fault framework for your complex structure. Grid horizon interpretation to create a true water-tight model that can be used in the model or high-quality structural maps. And shorten the time it takes to create a complex model by providing a cleaner interpretation and creating the structural framework in advance.
- Data management:Use the Seismic Survey Manager to effectively manage 2D and 3D seismic data within your Petrel project, improving your user experience when working with large regional areas, thousands of 2D lines with tens of thousands of traces, hundreds of kilometers and coordinate systems, and multiple 3D vintages and surveys.

2.3.4 Ocean for Petrel

Ocean is an application development framework with the capability to work across data domains. It provides services, components, and a common graphical user interface that enables efficient integration between applications. It allows application developers to interact with Ocean model-centric applications like Petrel. Ocean applications are loaded dynamically as .NET assemblies. These assemblies, the building blocks of Ocean, contain modules.

2.4 Matlab

MATLAB [21], short for Matrix Laboratory, is an environment developed by Mathworks, Inc. that facilitates matrix computations, numerical analysis and graphics viewing. The main reason for us to use Matlab in our work, is that MATLAB is often used by scientific researchers and engineers as it provides a high-level programming language that alleviates the users from low level programming details such as memory management and pointer handling. Complicated peripheral tasks, such as GUI creation, graph plotting, statistical analysis, and 2D image acquisition, can be readily handled by the wide variety of toolboxes available.

One more motivation to use Matlab in our work, is the potential speed advantage in computation and the facility for debugging, together with a considerable amount of established support [30].

There is an image processing toolkit supporting Matlab, but it is limited when compared with the range of techniques exposed in our work. taking in consideration that we have adjusted the different algorithms in our work to fit more with the Seismic data that we are focusing on.

Chapter 3

ALGORITHIMIC BACKGROUND

In this Chapter, we will present different algorithms that we build upon in our work. The original algorithms with mathematical background and the changes that we modified to fit our work will be explained separately in each section. This Chapter contains 2 main sections.

In Section 3.1, we will summarize the seeded region growing algorithm (Adams and Bishof), and then we will explain the inspired part in our work and the modification that we have done to be able to segment the salt bodies in seismic data. Sobel filter will be explained here as well, since we are using it combined with the seed growing in our algorithm.

In section 3.2, we will start by explaining what we mean by the term noise in seismic data. Moreover, a comparison between the different smoothing filters (median, adaptive median , mean and Gaussian) that we will be using in our work will be presented. Our focus in this section will be on mathematical background and the main principles that we used (The reader can skip the subsections that are familiar).

3.1 Seed growing

Automatic image segmentation is an essential process for most subsequent tasks such as image description, recognition, retrieval and object based extraction [33] which we will focus on in our work. Automatic image segmentation is considered as standard for realizing the object-based image coding and content-based image description and retrieval.

In our algorithm, we inspired the first step from the seeded-growing algorithms. In this step, we will define the borders of an object by defining a seed inside

the object by the user, and this seed will extend and grow (checking predefined conditions) to reach the borders.

In this section, we will define briefly the basic concepts that the seeded growing algorithm are built on. The last subsection will be the improved seeded growing algorithm that we modified and we will use in our work.

The general image segmentation problem involves the partitioning of a given image into a number of homogeneous regions according to a given critical [18]. Therefore, image segmentation can be considered as a pixel labeling process. All pixels that belong to the same homogeneous region are assigned the same label [20], the label in our case is going to be the Sobel values for the pixels.

3.1.1 Seeded region growing(Adams and Bischof)

The main concept that was presented by Adams and Bischof [1]is to segment an image into regions based on a set of Seeds, S1,S2,S3,..Sq,each step of Seed Region Growing (SRG) involves one added pixel to one of the seed sets. Furthermore, these initial seeds are replaced by the centroids of these generated homogeneous regions, R1,R2,..Rq, by linking the added pixels steps by steps. The pixels in the same region are labeled by the same symbol and the pixels in variant regions are labeled by different symbols. All these labeled pixels are called the allocated pixels, the rest are called the unallocated pixels. Let H be the set of all unallocated pixels which are adjacent to at least one of the labeled regions.

$$H = \left\{ (x,y) \notin \bigcup_{i=1}^{q} R_i | N(x,y) \cap \bigcup_{i=1}^{q} R_i \neq \phi \right\} \tag{3.1}$$

Where *N(x,y)* is the second-order neighborhood of the pixel(x,y) in 3*3 Matrix (we can figure here explaining the location of (x,y) in 3*3 matrix). For the unlabeled pixel $(x,y) \in H$,N (x,y) meets just on of the labeled image region R_i and define $\varphi(x,y) \in \{1,2,...,q\}$ to be that index in a way that $N(x,y) \cap R_{\varphi(x,y)} \neq \Phi$. $\delta(x,y,R)$ is defined as the difference between testing pixel at (x,y) and its adjacent labeled region R_i. $\delta(x,y,R_i)$ is calculated as:

$$\delta(x,y,R_i = |g(x,y) - g(X_i^c, Y_i^c)| \tag{3.2}$$

where $g(x,y)$ indicates the values of the three color components of the testing pixel (x,y), $g(X_i^c, Y_i^c)$ represnts the average values of three color components of the region R_i, with (X_i^c, Y_i^c) the centroid of R_i. If $N(x,y)$ meets two or more of the labeled regions, $\varphi(x,y)$ takes a value of i such that $N(x,y)$ meets R_i and $\delta(x,y,R_i)$ is minimized.

$$\varphi(x,y) = min_{(x,y)\in H} \{\delta(x,y,R_j) | j \in \{1,...,q\}\} \tag{3.3}$$

This seeded region growing procedure is repeated until all pixels in the image have been allocated to the corresponding regions. The definitions of Eqs. (1) and (3) ensure that the final partition of the image is divided into a set of regions as homogeneous as possible on the basis of the given constraints. SRG algorithm is robust, rapid and free of tuning parameters and it is also very attractive for semantic image segmentation. However, SRG algorithm also suffers from the problems of pixel sorting and automatic seed selection.

3.1.2 Modification to SRG

An advantage of SRG is that the high-level knowledge of semantic image components can be exploited by selecting the suitable seeds for growing more meaningful regions. This property is very attractive for content-based image applications [17], and this inspired us the idea of the labeling and the growing in a list gradually from SRG.

In our algorithm, the focus was how to manage the pixel labeling procedure more efficiently in such a way that can fit with the seismic images. The other point is the growing in our algorithm is not targeting the whole input image. It will start in a seed growing to reach the borders of the salt-object and stop there. This approach can save us a lot of calculations instead of run the algorithm on the whole image.

In addition to the performance , the estimate of region seeds or bad pixel sorting orders in SRG may result in an incorrect segmentation of an image [17]. The obvious way to improve the SRG technique is to provide a more effective pixel labeling technique and change the process of seed selection and add termination condition to stop it in a pre-defined manner than to run it based on the regions. In this section, we propose semi automatic seed growing algorithm for seismic images based on normalized Sobel filter in a boundary-oriented pixel labeling technique.

Using Sobel Filter as label in our growing algorithm

Sobel Filter is one of the standard image-processing edge detection techniques [20] to measure the luminosity of the onset of the RGB. We used it to produce an output reflecting the gradient detected across a three-point interval. As it is a first-derivative operator, it computes the rate of change across an edge. It is based on convolving the image with a small separable, and integer valued filter in horizontal and vertical direction.

The results that we got when we run Sobel filter show how 'abruptly' or 'smoothly' the image changes at that pixel, which can help usually in detecting the edges in inexpensive calculations for the images in general. However, one of the main characteristics of the seismic images is the noise which have changes that can show up as 'edges', the processing of the noise will be explained in the next section of this chapter.

Let A be the source image, and G_x and G_y are two images which at each point contain horizontal and vertical derivative approximations, the computations are as follow:

$$G_x = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} * A \tag{3.4}$$

$$G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} * A \tag{3.5}$$

where * donates the 2-dimensional convolution operation. The x-coordinate represents increasing the right-direction, while the y-coordination shows increasing the down-direction. For each pixel in the growing list in our algorithm, we will calculate the Sobel Value that equals the gradient approximations combined given the magnitude, using equation 3.6

$$G = \sqrt{G_x^2 + G_y^2} \tag{3.6}$$

3.2 Noise removal

Seismic data constantly consists of a signal and a noise component. As a general definition, we can say that any recorded energy which interferes with the desired signal can be considered as noise [40]. As a result this will affect our image, and more specificly the intensity of the noise can cause misinterpretation for the edges and boundaries .

The diversity of noise types often makes separation of signal and noise a challenging process. However, efficient noise attenuation and/or removal is important for high-quality image processing. From an industrial point of view it is also desirable that our algorithm should work on many types of similar noise without the need for time consuming parameter adjustments. Noise sources for the seismic data can be classified into two categories:

- noise comes from experimental errors. These errors involve any unexpected perturbation of the recording environment during data acquisition. A geophone can have malfunctions or the recording systems can have glitches creating erratic noise in the seismic record. Wind motion or cable vibrations can generate random noise. Outside factors, such as mammal activity and/or drilling rigs for marine acquisition might also contaminate seismic records. These noise sources create more coherent energy on the data and can be misinterpreted as true signal.
- noise comes from modeling uncertainties [4]. In seismic, modeling uncertainties occur when the physical description and parameterization of the

earth is incomplete. This incomplete description is motivated by the inherent complexity of wave field propagation in the subsurface. Therefore, the complex seismic signal is often separated into different propagation modes that are then easier to understand and use.

In our algorithm, we will focus on eliminating the coherent and incoherent noise. However, during the elimination of random noise we should keep in mind that no coherent components will be removed from our data. This step is critical because the result of it serves as the basis for all subsequent analyses. To detect the salt bodies we are trying to segment within a scene can be characterized by one similar feature which is the intensity in our case. In this section, we will represents filters to get image smoother and make the intensity of pixels specially of background pixels close each other. The suggested filters are the median, and the mean. It will follow comparisons between the effect for both of them in the results chapter.

3.2.1 The basic operation of digital image processing

To understand the median filter, the adaptive median filter and the mean filter that we will use in our work, we need to start with a short explanation for the basic operation in digital image processing. At each pixel in a digital image, we will define a window represents a neighborhood around that point (the size of this window will depend on the algorithm and the type of images that we are dealing with), analyze the values of the pixels in this neighborhood according to some algorithm to end up by replacing the original pixel's value with the one that our analysis performed on the pixels in our window. The neighborhood then moves successively over every pixel in the image, repeating the process.

3.2.2 Median filter

Following the basic description, the median filter will calculate the median value on the pixels in the defined window (neighboring pixels) and replace the pixel's value with the calculated median. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (In case that the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.) The median filter belongs to the class of edge preserving smoothing filters which are non-linear, which means that for two images A(x) and B(x): $Median\left[A\left(x\right)+B\left(x\right]\right. \neq Median\left[A\left(x\right)\right]+Median\left[B\left(x\right)\right]$

3.2.3 Mean filter

Like the median filter, the only difference simply is replacing the pixel value with the mean of neighboring pixel values instead of the median. As a convolution filter, it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. usually larger kernels can be used for more severe smoothing. In the same time, a small kernel can be applied more than once in order to produce a similar but not identical effect as a single pass with a large kernel.

3.2.4 Comparison between the median filter and the mean filter

In this subsection, we will present the main differences between the mean and the median filters on seismic data:

- The median filter is a non-linear tool, while the average filter is a linear one.
- In smooth, uniform areas of the image, the median and the mean will differ by very little. The median filter removes noise, while the mean filter just spreads it around evenly. The performance of median filter is particularly better for removing impulse noise than mean filter. Because the median is just the middle value of all the values of the pixels in the neighborhood, while in the mean filter we are defining a new value calculated by the other values in that window.
- The median has half the values in the neighborhood larger and half smaller, so it is considered as stronger "central indicator" than the mean filter in this perspective . In particular, the median is hardly affected by a small number of discrepant values among the pixels in the neighborhood. Consequently, median filtering is very effective at removing various kinds of noise.
- Although median filter is a useful non-linear image smoothing and enhancement technique. It also has some disadvantages. The median filter removes both the noise and the fine detail since it can't tell the difference between the two. Anything relatively small in size compared to the size of the neighborhood will have minimal affect on the value of the median, and will be filtered out. In other words, the median filter can't distinguish fine detail from noise.

In our implementation, we will try to test the effect of the smoothing filters on the seismic image specifically; since the noise in this case is considered a special case and the data set itself as well.

3.2.5 Adaptive median filter

The adaptive median filtering has been introduced as an improvement to the standard median filtering , as we explained before that the Median filter can detect the noise but in the same it can't differentiate between the fine details and the noise. So the main idea in the Adaptive Median Filter is to perform a spatial processing to determine which pixels in an image have been affected by impulse noise, and run the filter only in this pixels. The Adaptive Median Filter classifies pixels as noise by comparing each pixel in the image to its surrounding neighbor pixels. The size of the neighborhood is adjustable, as well as the threshold for the comparison. A pixel that is different from a majority of its neighbors, as well as being not structurally aligned with those pixels to which it is similar, is labeled as impulse noise. These noise pixels are then replaced by the median pixel value of the pixels in the neighborhood that have passed the noise labeling test. In other words, using this filter in our work is to remove impulse noise, smoothing of other noise, and to reduce distortion, like excessive thinning or thickening of object boundaries.

Let $x_{i,j}$, for $(i,j) \in A \equiv \{1,...,M\} \times \{1,...,N\}$, be the gray level of a true $M - by - N$ image x at pixel location (i,j), and $[s_{min}, s_{max}]$ be the dynamic range of x, $s_{min} \leq x_{i,j} \leq s_{max}$ for all $(i,j) \in A$. Denote by y a noisy image. In the classical salt-and-pepper impulse noise model [9], the observed gray level at pixel location (i,j) is given by

$$y_{i,j} = \begin{cases} s_{min}, & \text{with probability } p \\ s_{max}, & \text{with probability } q \\ x_{i,j}, & \text{with probability } 1-p-q \end{cases}$$

where $r = p + q$ defines the noise level. A breif review of the filter will follows.

Let $S^w_{i,j} = \{(k,l) : |k-i| \leq w \text{ and } |j-l| \leq w$ and let $w_{max} \times w_{max}$ be the maximum window size. The algorithm tries to identify the noise candidates $y_{i,j}$, and then replace each $y_{i,j}$ by the median of the pixels in $S^w_{i,j}$.

For each pixel location (i,j), do the following.

1. Initialize $w = 3$
2. Compute $s^{min,w}_{i,j}$, $s^{med,w}_{i,j}$ and $s^{max,w}_{i,j}$, which are the minimum, median, and maximum of the pixel values in $S^w_{i,j}$, respectively.
3. If $s^{min,w}_{i,j} < s^{med,w}_{i,j} < s^{max,w}_{i,j}$, then go to Step 5. Otherwise, set $w = w+2$.
4. If $w \leq w_{max}$, go to step2. Otherwise, we replace $y_{i,j}$ by $s^{med,w_{max}}_{i,j}$.
5. If $s^{min,w}_{i,j} < y_{i,j} < s^{max,w}_{i,j}$, then $y_{i,j}$ is not a noise candidate, else we replace $y_{i,j}$ by $s^{med,w}_{i,j}$.

The adaptive structure of the filter [9] ensures that most of the impulse noise are detected even at a high noise level provided that the window size is large

enough. Notice that the noise candidates are replaced by the median , while the remaining pixels are left unaltered.

3.2.6 Gaussian

Gaussian filtering has been studied in image processing and computer vision [7] [28]. Using Gaussian filter for noise suppression, the noise is smoothed out, at the same time the signal is also distorted [15]. The use of Gaussian filter as preprocessing for edge detection will also give rise to edge position displacement, edges vanishing, and phantom edges [27], [11]. We will look in this section to the mathematical background of the Gaussian filter in one or more dimensions [47] that we implemented later on in Matlab.

Gaussian Filter has been done as a convolution with samples of the required Gaussian (Eq. 3.7, 3.8), as repeated convolutions with a simpler filter such as a uniform filter (Eq. 3.9, 3.10), or as a recursive filtering with an approximation to the Gaussian that requires a complicated procedure to determine the filter coefficients.

The repeated convolution approach appeals to the central limit theorem which shows that, in the limit, repeated convolutions with an impulse response such as a simple uniform filter lead to an equivalent convolution with a Gaussian filter [44].

The discrete convolution with a sampled Gaussian is given per dimension by:

$$\begin{aligned} out[n] &= \sum_{k=-N_o}^{k=+N_o} in[k-n]g[k] \\ &= in[n] \otimes g[n] \end{aligned} \tag{3.7}$$

Where

$$\begin{aligned} g[n] = g(x|\sigma)|_{x=n} &= \frac{1}{\sqrt{2\pi\sigma}} exp(-x^2/2\sigma^2)|_{x=w} \\ & n = ..., -2, -2, 0, 1, 2, .., \end{aligned} \tag{3.8}$$

With σ real and N_o an integer. N_o is typically chosen as $N_o \approx \lceil 5\sigma \rceil$. At this value of N_o, the continuous Gaussian $g(x)$ is down by a factor of 3.7×10^{-6} from its value at $g(x = 0)$.

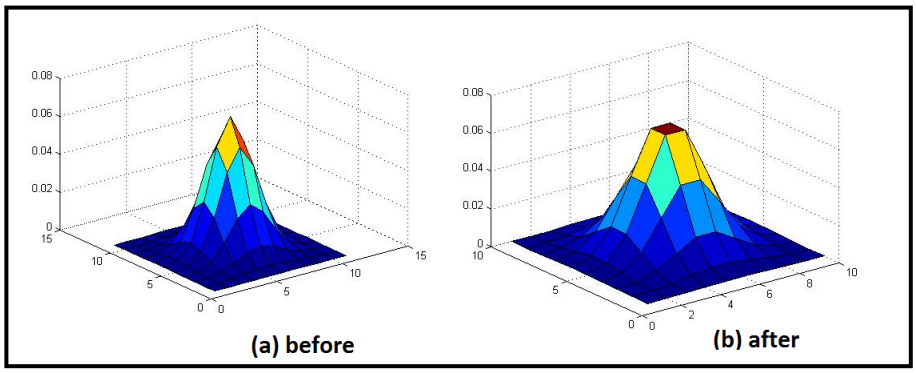
The implementation based upon repeated convolutions with a "square" kernel is given per dimension by:

$$out[n] = in[n] \otimes unif[n] \otimes unif[n] \otimes ... \otimes unif[n] \tag{3.9}$$

Where:

$$unif[n] = \begin{cases} 1/(2N_0 + 1), |N| \leq N_o, \\ 0, |N| < N_o \end{cases} \tag{3.10}$$

Figure 3.1 Effects of the Gaussian Filter in Matlab



We used three convolutions of a uniform filter to approximate a Gaussian. For a desired σ,this will then require $N_o \approx \lceil \sigma \rceil$. Only a limited number of Gaussian filters can be contructed in this way because we are constrained to use integer values of N_o (See Figure 3.1 Gaussian Filter in Matlab).

Chapter 4

METHODOLOGY AND IMPLEMENTATION

In this chapter, we will discuss the Methodology in more details than the previous chapter, and some of the implementation particulars will be mentioned here referring to Matllab as our environment to test. This chapter will be divided into five sections.

In section 4.1, we will have a short description of our method. The sections that will follow will explain in more details the different steps in our method.

In section 4.2, we will talk about the input data to our algorithm. The whole process since the moment that we got the original seismic data, going by the different attributes that can help us to get better results. In addition to the synthetic data that we used to test the different ideas that we tried, and the relationship between these simple shapes and the real data.

In section 4.3, we will talk about the seed growing algorithm and the use of normalized Sobel filter.

In section 4.4, we will talk about the noise in our data, and the different filters that can be used to remove that kind of noise.

In section 4.5, we will talk about the disconnected areas when detecting the salt bodies, and the different methods and implementation that we used to overcome this.

In section 4.6, we will represents two different approaches to detect our starting seeds automatically , instead of the user define them manually.

All the results related to these different methods and algorithms will be found in the next chapter following the same order with the same titles. So the reader can move smoothly between the two chapters.

4.1 Overall View

In this section we will have a short description of the whole process that our algorithms will cover (See Figure 4.1). We will divide the whole process into 5 stages:

1. Define the start seed: In this stage we will have processed Post-Stack Seismic Data Volume Attribute Cube as input to our algorithm (More details in section 4.2). We suggested that the definition of the seed points can be done manually base on interaction from the user or automatic definition.
2. Growing Phase: This includes the first extension to the start seeds. The seed will extend to 8 pixels (Points), more details will follow in section 4.3.
3. Noise Removal Phase: In this phase, we will evaluate the chaotic texture of the neighborhood, and select a smoothing filter accordingly, more details will follow in section 4.4.
4. Edge detection Phase: In this phase, we will detect the salt edges after we removed/reduced the noise by using Sobel values, more details will follow in section 4.3.
5. Estimate the discontinuous edges: by defining termination condition that can stop the growing around the discontinuous edges, more details will follow in section 4.5 The Growing phase, noise removal phase, edge detection and estimate the discontinuous edges will be repeated until the salt body is detected.

4.2 Input Data

In this section , we will explain our choice in the synthetic data that we tested our algorithms on. In addition to the input data (Seismic data set)and the attributes that we used in Petrel to process before running our algorithm.

4.2.1 Why are we considering our synthetic data as a circle?

We will consider our synthetic data for the salt body as circles, based on our understanding of the salt tectonics structure. The gravidity will push the structures with high density down, while pushing the salt with the low density up along the faults, traveling up to the surface like below (See Figures 4.2 and 4.3). Or it can go up by the faults and expand.

Figure 4.1 Diagram of Automated Seeded Growing Method for Salt Body Delineation

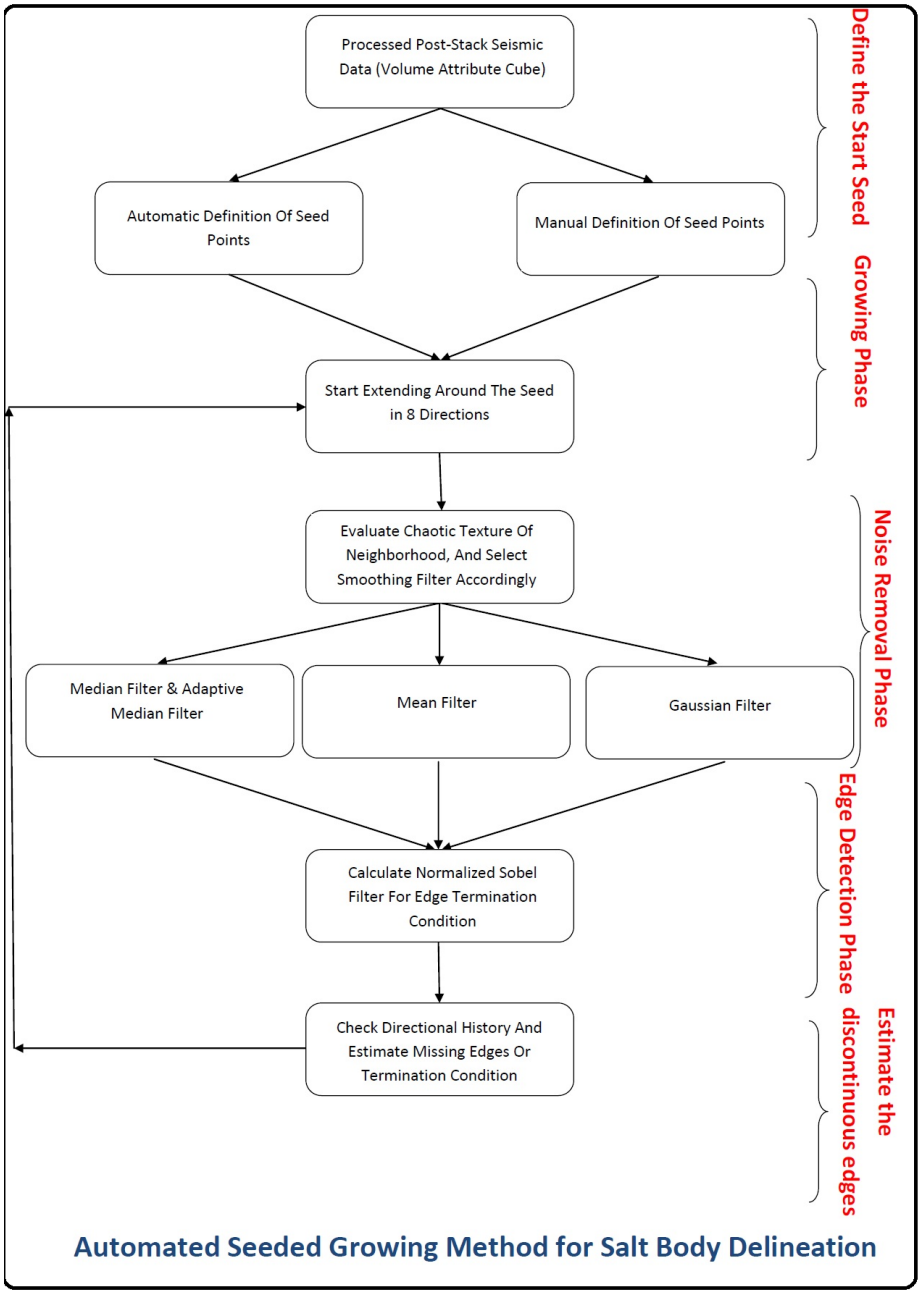
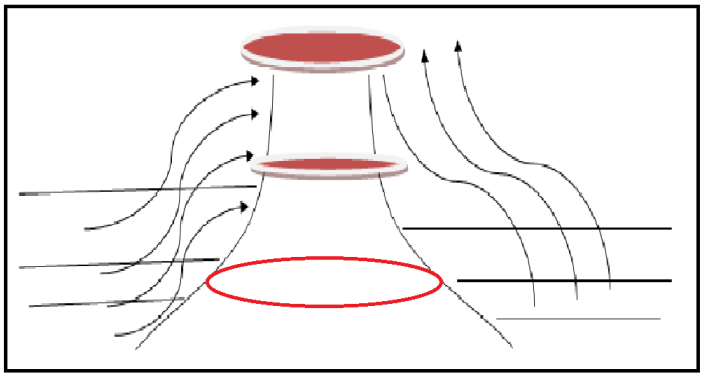


Figure 4.2 Salt Structure Inline view



The gravidity will push the structures with high density down, while pushing the salt with the low density up along the faults, traveling up the surface like a below. Or it can go up by the faults and expand.

Figure 4.3 Salt Structure [24]

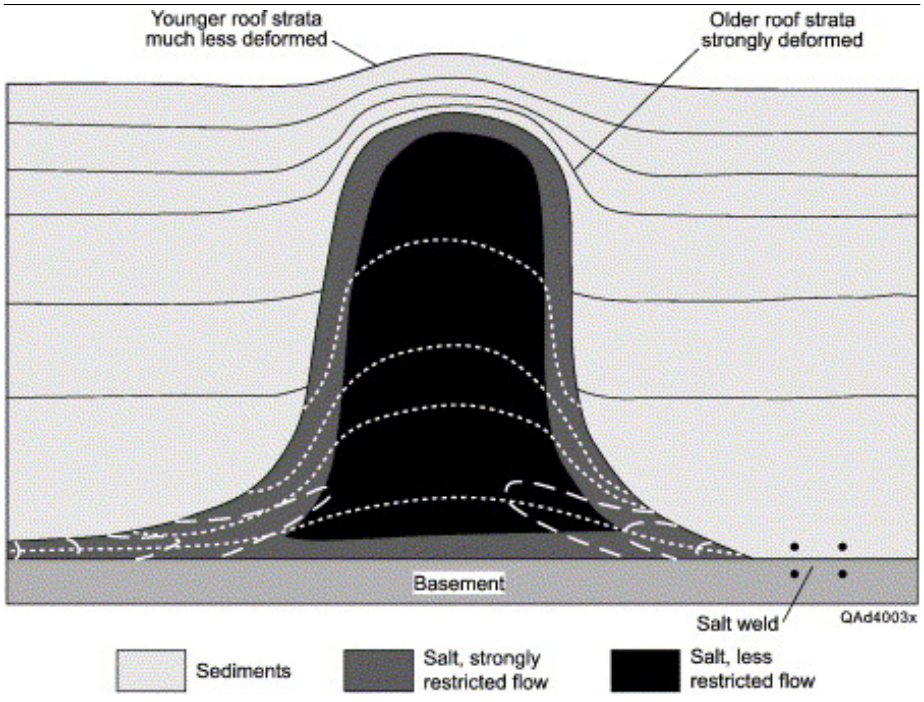


Figure 4.4 Inline, Crossline, timeline in Seismic. Schlumburger Oilfield Glossary

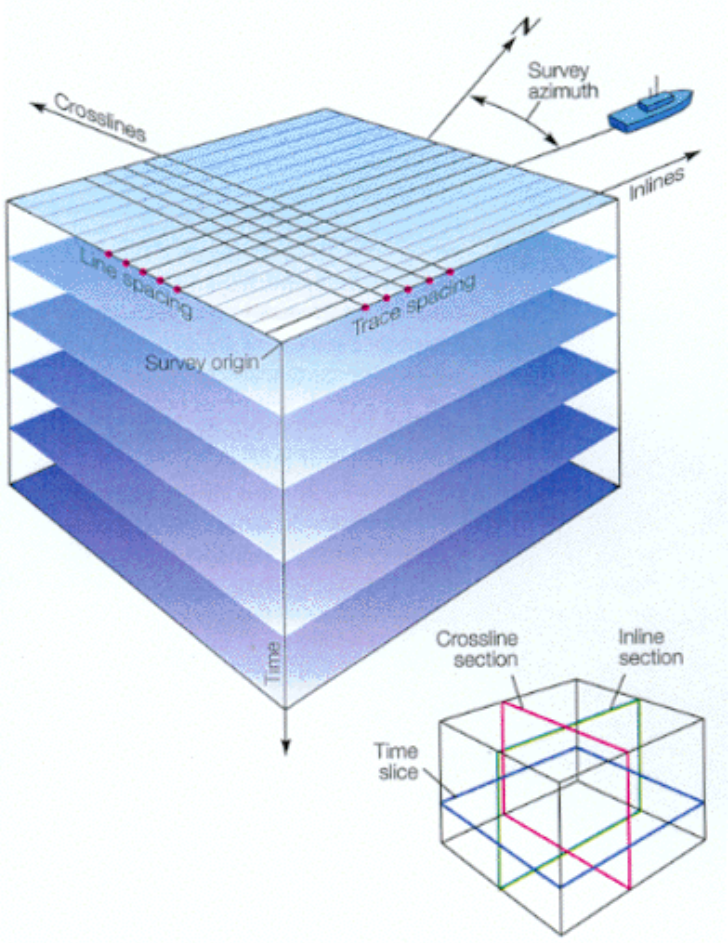
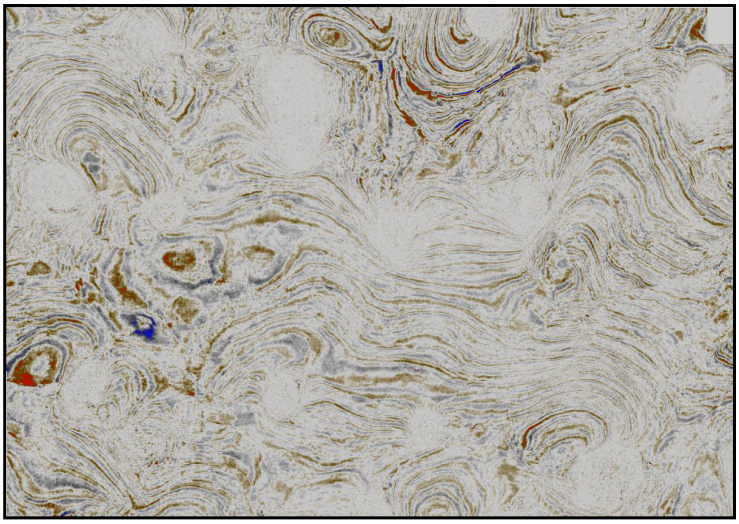


Figure 4.5 Original Seismic (Depth-Slice) Salt delineation- courtesy of WesternGeco



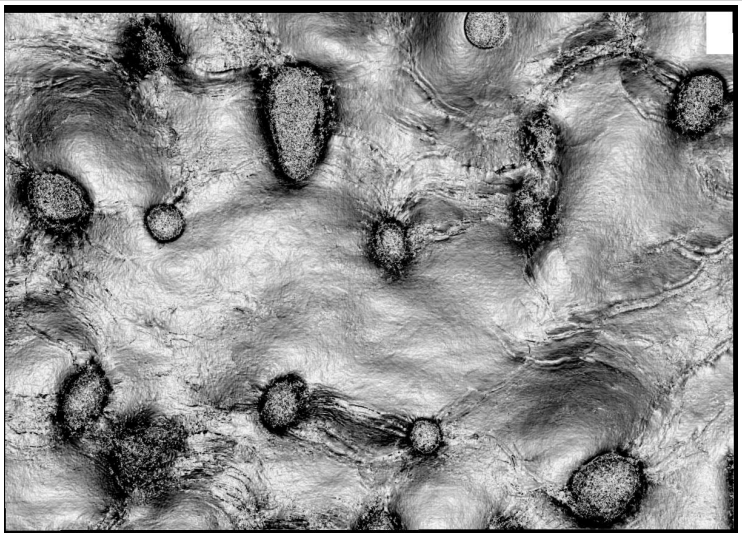
Looking into the salt structure in depth slice, inline slice, or cross slice (See Figure 4.4)will give us the shape of the circle, or combined circles.

The real input data for our algorithm will be filtered using the seismic attributes in Petrel. The original seismic (depth-slice) will look like Figure 4.5. We considered this seismic as a sample to test our algorithm we applied it to the following attributes:

4.2.2 Dip estimation attribute "Dip Illumination "

In this attribute [2], an accurate dip estimation can be viewed as changing the light source either directional along the azimuth or perpendicular to the plane. This attribute includes dip scan method for estimation and Hill Climb scan method for enhancement. Dip scan is calculating the maximum local correlation of every Δt pixels and if Δt pixels is less than 1, then it will interpolate values between the samples to estimate true correlation peak. In Hill climb scan, it will start at dip 0, determine scan direction, scan while correlation increase and stop when correlation decreases again i.e reaches a local peak. For Dip estimation, a weighted estimate 5*5 trace neighborhood was used where more weight were given to central traces. Figures 4.6 and 4.7 show the result of this attribute. The search for the maxima is a discrete search using a greedy hill climb algorithm. It searches with a fixed step until it finds a correlation maximum position (pos_{max})

Figure 4.6 Dip Illumination (moon map)



is estimated using Newton's method 4.1:

$$pos_{max} = pos_m - \frac{\grave{f}pos_m}{\grave{f}pos_m} \tag{4.1}$$

Given all trace to neighbor trace dips for all samples we estimate the dip at each sample in a volume by using the averaged and weighted sum of a 5*5 neighborhood around that sample. The main goal of using these display methods is to give us greater insight in the structural features, where we can get more information than we usually get from the edge attributes.

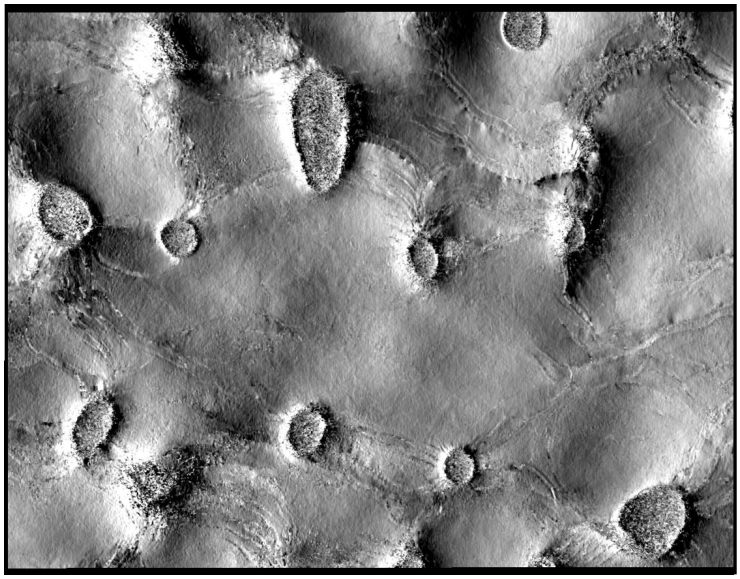
4.2.3 Salt detecting attribute "Amplitude contrast"

Amplitude contrast is a patented Sobel based attribute (Aqrawi and Boe, 2011)[3]. It is basically a computation of the amplitude derivatives between neighboring traces , the weight of the non diagonal neighbors is twice as the diagonal. The calculated differences are then normalized to help minimizing the horizontal artifacts of the vertical differences to hide the stratigraphy. The final value is calculated using Equations 4.2,4.3,4.4,4.5, Where S_x,S_y and S_z are the weighting operators in the corresponding dimensions.

$$Gradient_x = S_x * Input \tag{4.2}$$

$$Gradient_y = S_y * Input \tag{4.3}$$

Figure 4.7 Directional Dip Illumination



$$Gradient_z = S_z * Input \tag{4.4}$$

$$Result = \sqrt{Gradient_x^2 + Gradient_y^2 + Gradient_z^2} \tag{4.5}$$

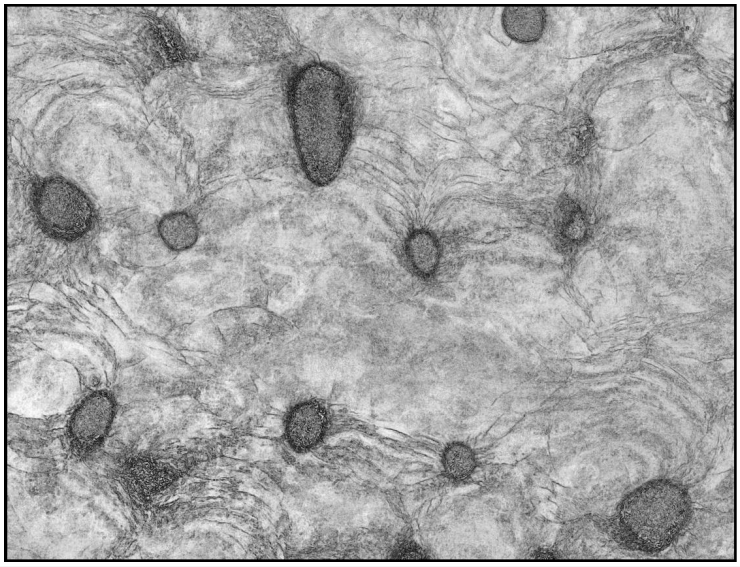
The dimension weighting is to weight the 3 dimensions differently, a fraction of the value of the horizontal dimensions (time or depth). Figure 4.8 shows the result of attribute Amplitude Conrast on the original Seismic 4.5

4.2.4 Edge enhancement attribute "Edge evidence"

We used a statistical edge enhancement method[38] was inspired by a windowed Randon(Randon 1917) [16] or Hough (Hough 1959) [32]. This method is searching for local edges in 2D sections of a seismic cube and was developed to include the 3D by (Boe and Aqrawi 2012) [38]. The method evaluates evidence locally at each pixel in an image, e.g. whether there is evidence of a line through that specific pixel p. Given an operator radius r, and an imaginary line going through p at an angel θ, the evidence for a line running through pixel p is evaluated by comparing the values on the imaginary line with the surrounding values. For each candidate line given by the angle θ, it is possible to use the sum of pixel values as the evidence measure.

For each pixel in the image, we will perform the test for many different angles passing through it. Then we choose the most significant evidence from all can-

Figure 4.8 Dip Guided Amplitude



didate lines and calculating the negative logarithm of the P-Value. The P-value here means the probability of observing something that may look like a line in a random noise image. When we calculate the negative logarithm of the P-values, the most significant evidence of lines will produce the highest output values. The main goal to use this method in our data is to filter out the high energy speckle noise and also to give a response that is independent of the input amplitude.

4.3 Seed growing with Sobel Values

In this work, we are proposing a method to detect the boundaries of the salt bodies, the basic idea will start by interaction from the user to define a point inside the body that we want to extract, this point will be the seed that will grow in all directions (it will extend 8 cells around the chosen cell. See Figure 4.9). Every time we want to add new cells to our list we will have a termination condition will be checked, and then decide if any more extension or termination is needed at that point. We will compare different termination conditions and represent the different cases when using diverse filters later on in this section.

Since we are working with seismic data, that is really well known as noisy data, some preprocessing will be implied to remove that noise. As the goal of this work is to define a method can be part of a work flow to integrate it later on in Petrel Software. The input data will be processed by before mentioned filters in Petrel, before running our algorithm ((More details in section 4.2)). However,

Figure 4.9 Seed Growing by extending 8 cell around the chosen cell

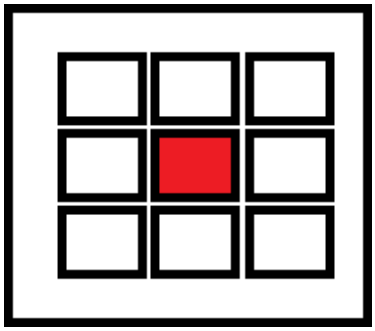
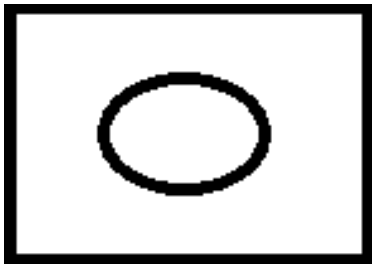


Figure 4.10 Basic Case- To check detecting the borders



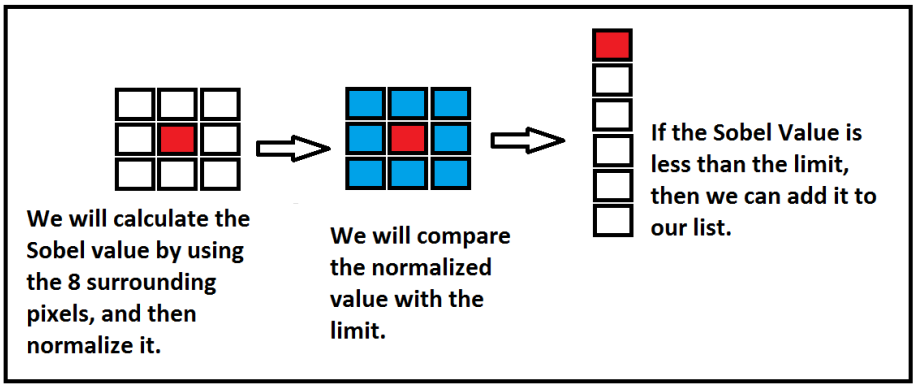
we preferred to start with synthetic data to test the different cases in Matlab and after that we tested it directly by using seismic samples.

We started with the basic case, where we define a closed body that has the shape of a circle Figure 4.10, and the user has to define a point inside that circle, this seed will be growing with a predefined color and stop automatically when it reaches the borders. At the first round, the seed will extend 8 cells (the cells in our case will be defined as an object called MyPoint. MyPoint object will have the X,Y,Z coordinates,the direction of this point related to the seed in addition to a flag defining if this cell was visited or not to avoid adding the same cells more than once which can cause an infinite loop). We will add these points to a List, and then we will start checking the Sobel value for each point separately.

Sobel value are typically the computation of the derivatives between neighboring traces. In addition to the classical Sobel calculations, we have added the normalization. The normalization helps to avoid discrimination between high and low amplitudes and noisy data. We have done that by using the absolute sum of the values in the sub cube calculations are performed on, to look directly at the change in a sub-cub normalized by its sum.

After getting the Sobel value, we will have the condition which will check if Sobel value is high then we know that we detect a change in that point which

Figure 4.11 Sobel- Seed Growing



means "It can be an edge", then the growing process at that point will stop and it will keep the original color to that point (Black). While if Sobel value was low, then it is probably not an edge and we should color it and grow around this point again to add more 8 cells to our list after making sure that the added cells are not repeated by checking the visiting flag.

This process will continue until we check all the points in the list. As the reader can see in Algorithim 1, the list is growing step by step, so no extra calculation is used. Which means that this simple concept will not be time or memory consuming, as if we will run our algorithm without a seed to start with. Figure 4.11 shows how one pixel will depend on 8 surrounding pixels, to define if it will join the list and extend more or define it as an edge and stop there, and then we will check the next element in our List.

After running our algorithm on synthetic data, we tried to run it on processed seismic data see Figure 4.12

4.4 Seed growing with Noise

One of the main challenges in working with seismic data is the noise, so we will add some noise to our synthetic data (consider it as salt and pepper noise for simplicity), and run Algorithm 1 Seed growing-Sobel and check the result. See figure 5.6.

The circle represents our input data with artificial noise, we will start by running seed-growing based on the Sobel filter only, Check the Results Chapter(Section 5.2).Thinking in a very simple way, we will consider that the noisy pixels in the image are very different in color or intensity from their surrounding pixels; the defining characteristic is that the value of a noisy pixel bares no relation to the

Figure 4.12 Our Seismic Sample to test our algorithm

Algorithm 1 Seed growing-Sobel

Require: Defined point by the user
The seed will extend 8 more points around it.
The seed will be added to my List in the first round
while There is a point i in the List not visited **do**
for all $i = 1 \rightarrow$ the size of the List **do**
Calculate Sobel Value at Point i
if Sobel Value at point $i \leq Limit$ **then**
This is not a border, Color it with the new color.
Extend 8 cells around the Point i
for all $j = 1 \rightarrow 8$ **do**
if Point j is not visited **then**
Add Point j to the List
Update the size of the list
end if
end for
else
Sobel Value at point $i < Limit$ A Border was detected, Keep the original colorSTOP Extending around this point
end if
Mark The point as VISITED Point
end for
end while

Figure 4.13 Seed growing with Salt and Pepper Noise

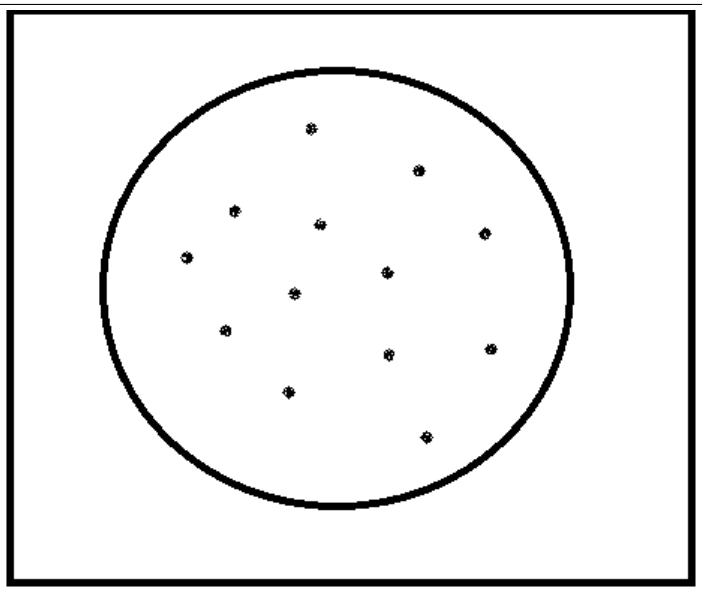
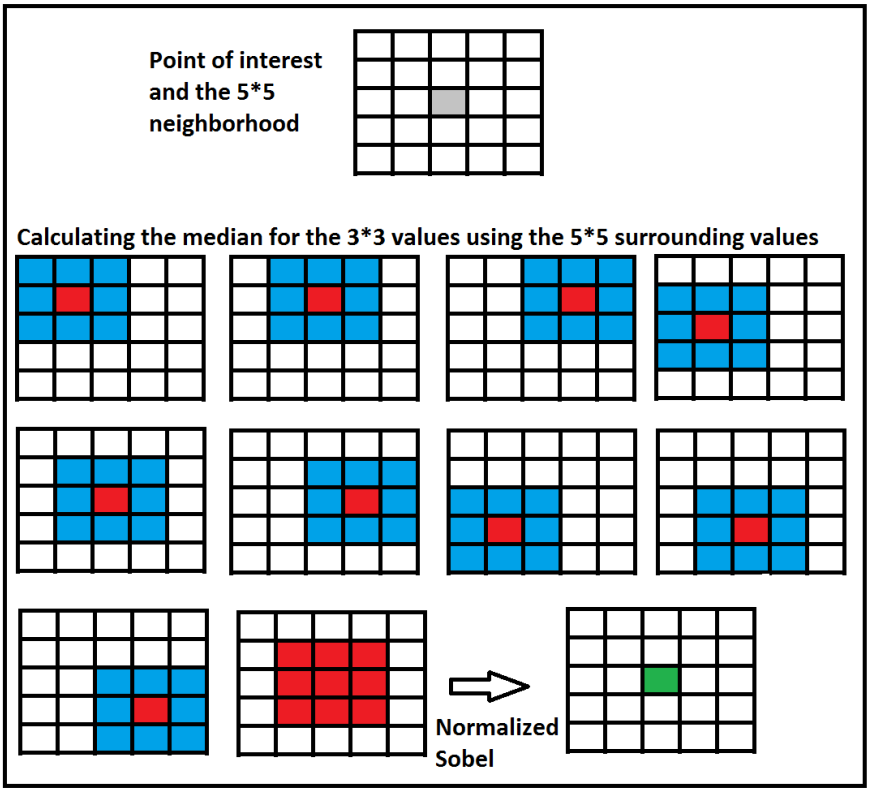


Figure 4.14 Median Sobel Combination



color of surrounding pixels. The change in the colors will cause high differences in the derivatives.

However, we can define the noise from the edges by looking to the surrounding pixels. So we will test different filters with different scales, to remove the noise before running the Sobel filter, and it has to be seed growing as well. To keep the quality of our data, we will avoid running the noise filter over all the image before start running our algorithm, and just run it while we are growing step by step (See algorithm 2). We are going to test different filters to remove the noise such as Median, adaptive median,mean and Gaussian filters. Afterward we will see which one fits best with the seismic data. To test this combination between filters to remove the noise, the output of the filters will be the input to the Sobel filter, which will be extended based on a condition as a Seed growing manner we will follow these basic steps in Figure 4.14.

To calculate normalized Sobel value for one pixel, we need to calculate noise filter for the 3*3 values. This will require 5*5 surrounding neighboring pixels. In Figure 4.14, we will consider that the noise filter will be the median filter,

and show example.

Algorithm 2 Seed growing-Sobel and Noise Filter

Require: Defined point by the user
  The seed will extend 8 more points around it.
  The seed will be added to my List in the first round
  while There is a point i in the List not visited **do**
    for all $i = 1 \rightarrow$ the size of the List **do**
      Remove the noise by checking 5*5 neighbors
      Calculate Normalized Sobel Value at Point i using the new values from the noise filter
      if Sobel Value at point $i \leq Limit$ **then**
        This is not a border, Color it with the new color.
        Extend 8 cells around the Point i
        for all $j = 1 \rightarrow 8$ **do**
          if Point j is not visited **then**
            Add Point j to the List
            Update the size of the list
          end if
        end for
      else
        Sobel Value at point $i > Limit$ A Border was detected, Keep the original colorSTOP Extending around this point
      end if
      Mark The point as VISITED Point
    end for
  end while

To select a noise reduction algorithm, we will take in consideration the tradeoffs between:

- The available computer power and time, especially that the seismic data has huge size usually.
- The real details that we can accept to sacrifice when remove the noise, the accuracy in the filters to decide whether the variations in the image are noise or not.
- The characteristics of the noise and the details in the image, in the case of the seismic data.

We tested 3 different filter to reduce the noise in our data, adjusting the seed growing concept in the same way as we explained earlier.

4.4.1 Median filter

we implemented the Median filter by defining our scale, sort the values in our range in a list and then pick the median of these values and replace this value to the pixel that we are processing.We tried different scales, 5*5 neighbors, 7*7 neighbors, 9*9 neighbors. Keeping in our mind the tradeoffs between the time that we are using, the noise remove and keeping as we can all the details in our data while we are filtering.

4.4.2 Adaptive median filter

In the Adaptive Median Filter, we performed a spatial processing to determine which pixels in our image have been affected by impulse noise by checking this condition and then label it using a flag, and run the filter only in these pixels with flags. We classified pixels as noise by comparing each pixel in the image to its surrounding neighbor pixels. The size of the neighborhood is adjustable (We tried different cases, check the Results Chapter), as well as the threshold for the comparison.

A pixel that is different from a majority of its neighbors, as well as being not structurally aligned with those pixels to which it is similar, is labeled as impulse noise. These noise pixels are then replaced by the median pixel value of the pixels in the neighborhood that have passed the noise labeling test.

4.4.3 Mean filter

The same concept as the Median filter, we defined our scale,we defined our scale to the neighboring pixels that we want to check, and save them in a list where we calculated the mean for all the values in the list. Then the pixel value will be replaced by the mean. different Scales were tested to check 5*5, 7*7, 9*9. When the scale goes bigger ,then the filtering level will be affected with wider area in the image.

Here we followed the same pattern as the previous sections, we run the filter on the original values and after filtering the seeded area pixels, we run Sobel to detect the edge. If we want to expand more, we will start again with the filtering part for the new pixels and then Sobel again.

4.4.4 Gaussian filter

The same concept as the previous filters , we defined our scale (the neighboring pixels that we want to check), and saved them in a list where we calculated the Gaussian equation that we mentioned earlier in Section 3.2.6 for all the values in the list. Then the pixel value will be replaced by the Gaussian value.Then

we run the Sobel filter to detect the edges. If we want to expand more, we will start again with the Gaussian filtering part for the new pixels and then Sobel again.

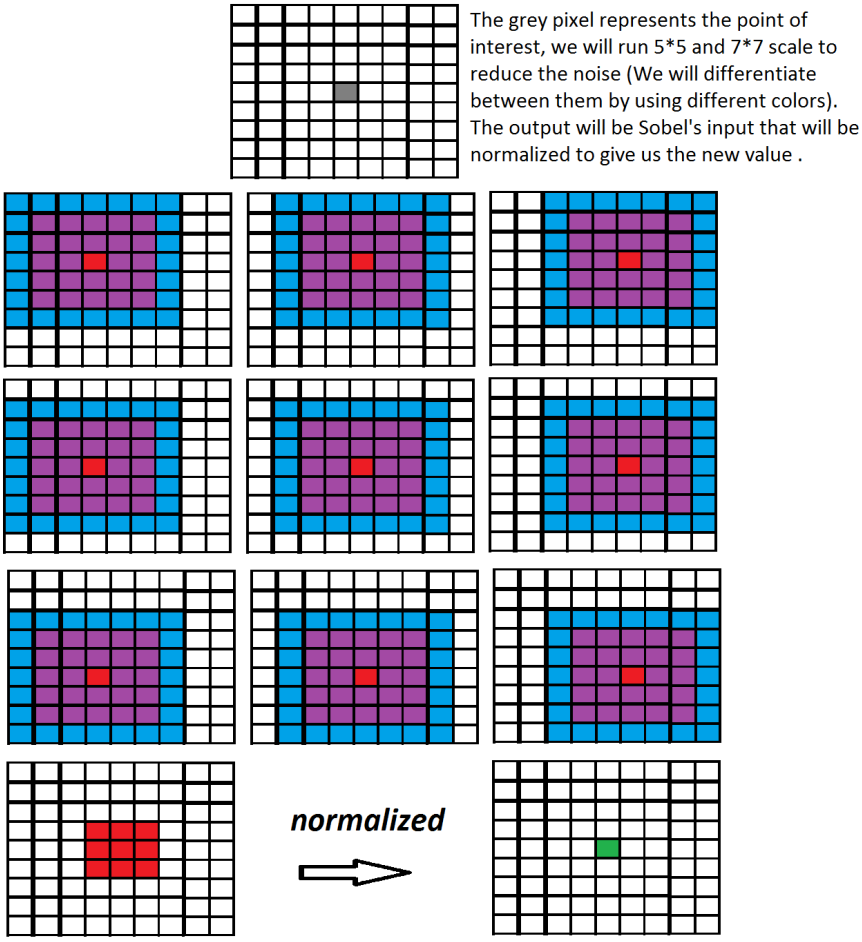
4.4.5 Hybrid combination of smoothing methods

After studying and implementing different smoothing filters, We have also taken into account when to use the various filters for best results and suggest to switch between them, in real time, given an analysis of how chaotic the considered neighborhood is. This means that we are using a hybrid combination of smoothing methods, and our alternating criteria, in this case, is the amount of chaos detected.

So before running any smoothing filter on our pixel, we will calculate the variance value for the neighboring pixels. Based on the calculated value, we will choose which filter to use. we will have two main cases:

- If the variance value is high: this means that the values around the targeted pixel are chaotic and that can be caused by noisy pixels in my scale. So using the median-Or the adaptive median filters will give us better results in smoothing the noise instead of spreading the noise around the pixels in my scale.
- If the variance value is low: this means that the values around the targeted pixel are close. Using the mean or Gaussian filters can give better and accurate results than using the median here,because we will average the values to remove the noise while preserving the nature of the data.

Figure 4.15 Smoothing Filter with 7*7 Scale and 9*9 Scale



4.5 Disconnected areas

In this section, we will discuss the disconnected areas in the salt bodies and suggest different algorithms, that we can shape the salt bodies when some parts are not continuous. In other words, we will try to estimate the discontinuous areas based on our understanding to the Seismic surveys.

the seismic interpretation of seismic surveys is highly dependent on the geological context. (More details, check section 2.1.2) we have quick explanation about the shape of the salt bodies, then we will present the different filters that we tested with seismic data and especially with the salt tectonics.

4.5.1 Input data

In this section, we will discuss our methods on the salt bodies that have discontinuous parts, and see how our algorithm will estimate these parts. The growing pattern should stop at some moment and define the missing edges in the salt body. The Synthetic data is going to be the same like the one that we used before in addition to some discontinuous parts that we will add (See Figure 4.16) In addition to the synthetic data, we chose a sample from Seismic data with disconnected salt structure (the data set from the gulf of Mexico). We did some filtering using Petrel and then we used it to test our methods. In Figure 4.17, we can see the original seismic Inline (Side view). The Salt is pushing from both sides (left and right) to the top. By running Amplitude Contrast, we can notice how the dark areas show us better the location and the structure of the salt. In Figures (4.18,4.19,4.20 and4.21), We run the Dip illumination and the Instantaneous dip Inline (side view). The black lines represent the flanks (the borders of the salt body). It's just a steep dipping feature, that is a reflector. The horizontal lines that we see in seismic are bending at a steep angel.

The sample that we will test in Matllab is in Figure 4.22 with the red circles that we will estimate and stop our growing algorithm automatically there.

Figure 4.22 shows that the salt bodies are not always continuous perfectly as circle. It will have some areas that are disconnected (In the red circles).

4.5.2 Directional history for the seeding points

As we explained earlier, another unfortunate characteristic in salt data is that it may have discontinuous edges and hence a growing algorithm will grow beyond the salt structure. Our attempt to address this involves estimating the boundary along discontinuous parts of the structure and terminate accordingly. There are several approaches that could be used for the termination estimation, such as using the double derivative and edge enhancement techniques. We have focused

Figure 4.16 Our Input data

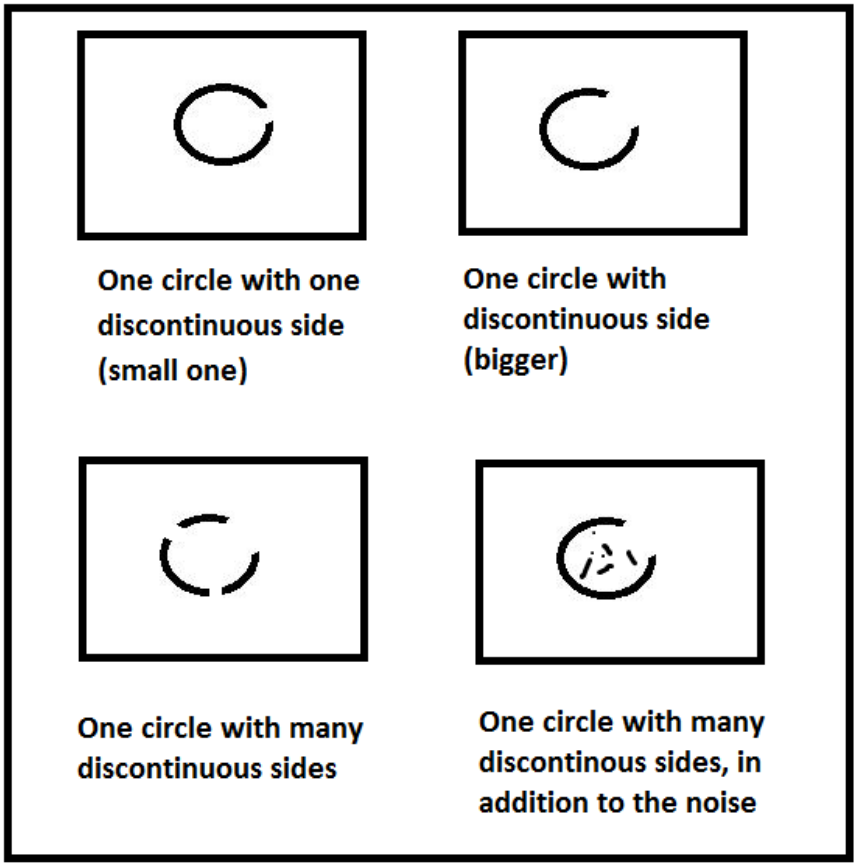
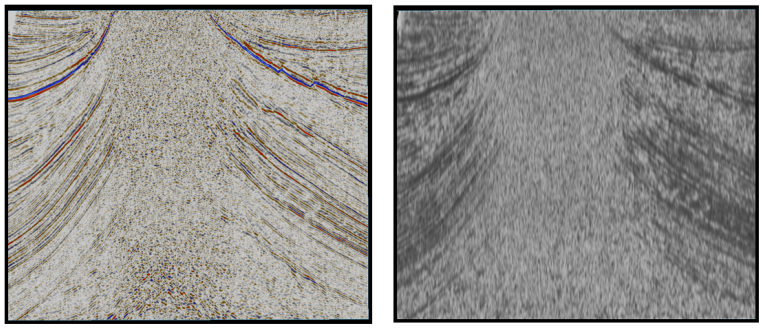


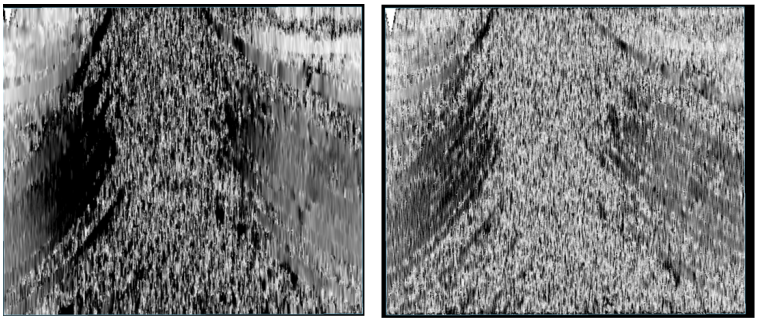
Figure 4.17 Amplitude Contrast



Original Seismic: Inline (side view)

Amplitude Contrast: Inline (side view)

Figure 4.18 Dip illumination and Instantaneous dip



Dip illumination: Inline (Side view)

Instanteous dip: Inline (Side view)

Figure 4.19 Position of horizontal and vertical slices

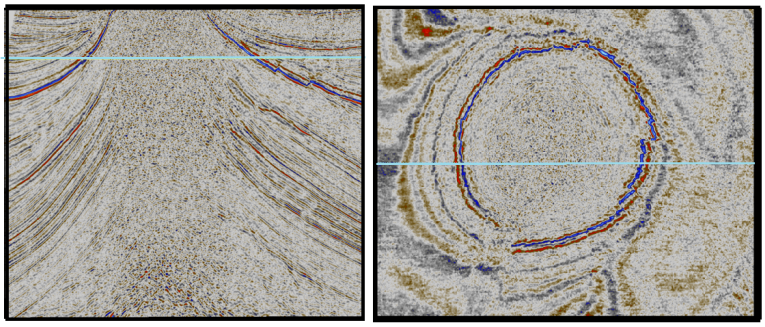


Figure 4.20 Amplitude Contrast

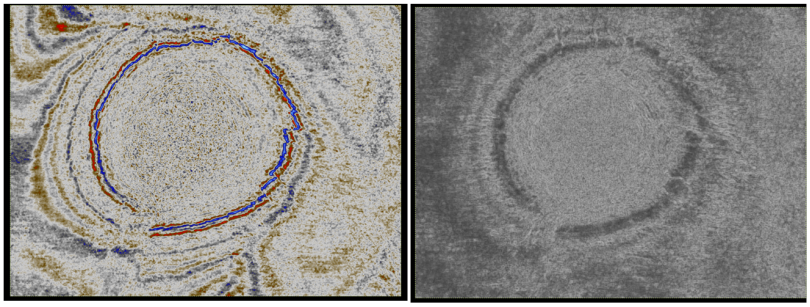


Figure 4.21 Dip Illumination and Instantaneous Dip

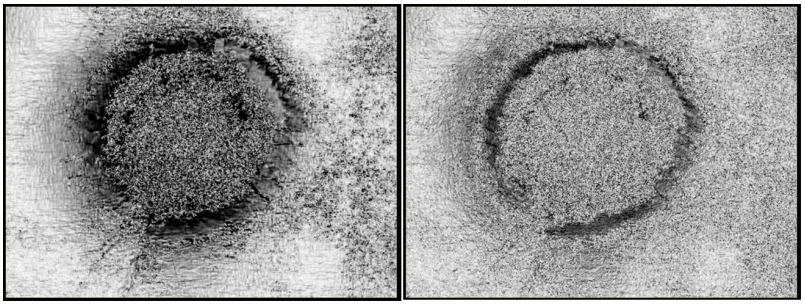
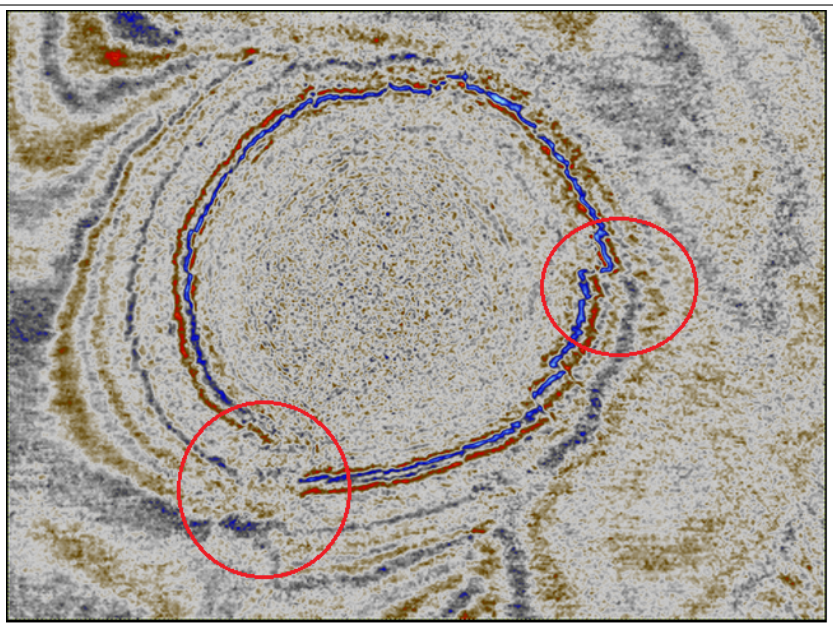


Figure 4.22 Disconnected salt structure

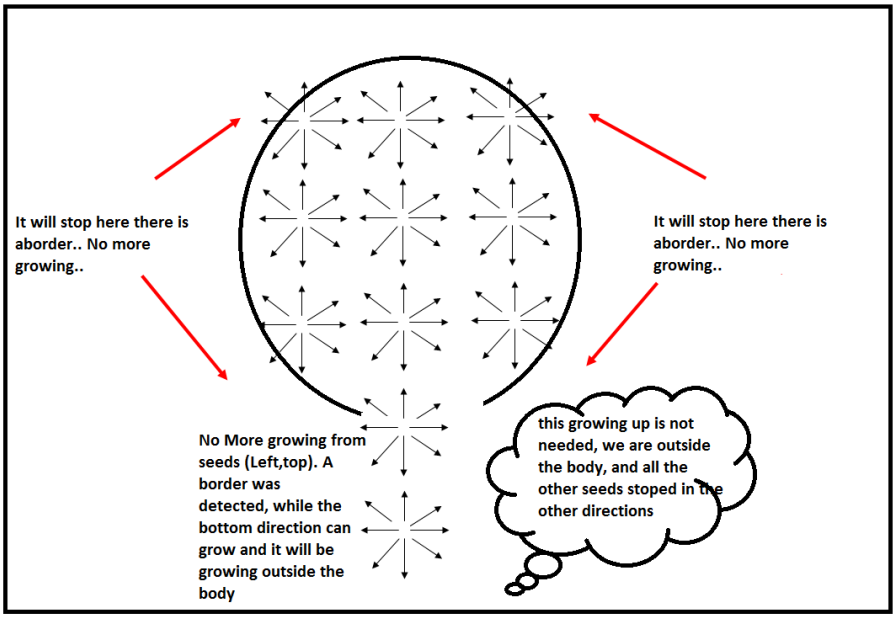


The red circles show the discontinuous areas

on evaluating the growing directions of our seed points and terminate based on unbalanced behavior.

In our Seed growing algorithm, the seeds will keep growing from the disconnected areas (See Figure 4.23). This is because Sobel values there are going to be small and no borders will be detected. The growing will be outside the bodies. So we need to add to our algorithm some termination condition to check while the seeds are growing that the behavior of the growing is normal (with our prior knowledge to the expected shapes for the salt bodies). One way to fix this, is

Figure 4.23 Growing outside the Salt body



to check the direction history. When our seeds are growing, they are extending as 8 cells around the seed. This will help us to define the directions that we are growing through. And then keeping this history will show us if the ratio while we are extending in one direction is not normal. This can be done by comparing it to the other directions. A very important point needs to be considered, is the defined seed by the user (the first seed we will start from) should be centered inside the detected object.

In algorithm 3, we added into the while condition the termination condition that will check the connectivity and terminate the growing. Here we will explain "The Connectivity Function" in our algorithm, and how we implemented it in Matlab. When we are defining a seed , we will have 8 more seeds that are growing around this seed, we will define a variable in the Object Point (Our seed) and call it direction and there we will save the direction for this seed in

Algorithm 3 Seed growing-Estimate the missing part Of the salt

Require: Defined point by the user
The seed will extend 8 more points around it.
The seed will be added to my List in the first round
while There is a point i in the List not visited **AND Check that the Connectivity Function is True** **do**
for all $i = 1 \rightarrow$ the size of the List **do**
Calculate Sobel Value at Point i
if Sobel Value at point $i \leq Limit$ **then**
This is not a border, Color it with the new color.
Extend 8 cells around the Point i
for all $j = 1 \rightarrow 8$ **do**
if Point j is not visited **then**
Add Point j to the List
Update the size of the list
end if
end for
else
Sobel Value at point $i \succ Limit$ A Border was detected, Keep the original colorSTOP Extending around this point
end if
Mark The point as VISITED Point
end for
end while

comparing to its parent. Every time we are extending in the same direction, we will save the directions of the new the seeds and keep it in a list called direction List. In the connectivity function we can define the conditions related to the direction (Different conditions can be applied here, like comparing the different directions with each other, or comparing it with the total list).

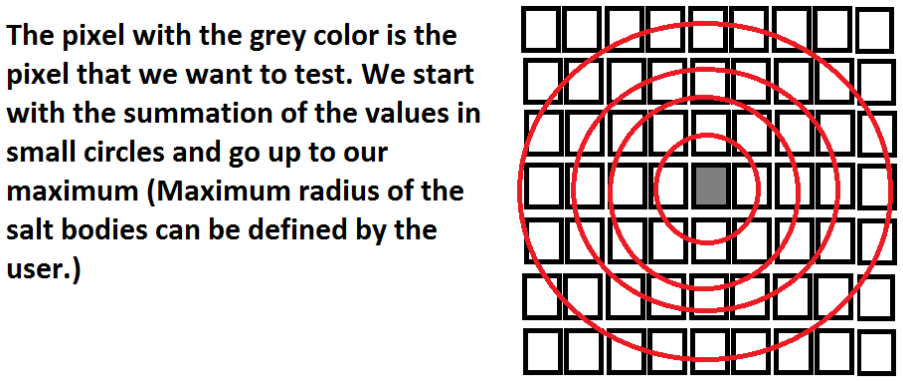
The expected behavior that the growing algorithm will run normally, until we have some disconnected area. In this point, it will be extending a little bit to estimate the missing edges and then it will stop automatically.

4.6 Automatic detection of the seeds

One of the features that we can implement and add to our algorithm to make it more attractive and easy to use, is the automatic detection of the seeds. We suggest here two different approaches to detect the seeds automatically.

4.6.1 Check if the targeted pixel is a seed point

Figure 4.24 Check if this pixel is a seed.



In this approach, we will start checking all the pixels in my image one by one to check if a pixel is a seed or not. The test is simple, we will find the summation of the neighboring pixels in circular base, If the summation has high value, then the tested pixel is a seed. Otherwise, we will increase our scale to check the summation of neighboring pixels in circular base with higher radius, we will keep this check until we reached the maximum radius that we expect for our salt bodies. After that we will move into the next pixel to repeat the same test(See Figure4.24).

Figure 4.24 shows the pixel with the grey color as the pixel that we want to test. We start with the summation of the values in small circles and go up to our maximum (maximum radius of the salt bodies can be defined by the user).

Algorithm 4 Automatic detection of the seeds

Require: The input will be the image (the seismic data)
  for all $i = 1 \rightarrow ImageLength$ **do**
    for all $j = 1 \rightarrow ImageWidth$ **do**
      for all $k = 1 \rightarrow ImageDepth$ **do**
        Calculate the sum of the neighboring pixels (circular neighbors)around (i,j,k)
        if The Sum is $\gtrless$ Limit **then**
          The tested pixel is a seed and we will mark it
        else
          Increase the scale of the radius and check the sum again
        end if
      end for
    end for
  end for

4.6.2 Define my edges and then calculate the seed based on

In this approach, we will start by smoothing the image. The next step is to check the values of all the pixels in my image. Then, define the pixels with the high values in a list and then we will sort the list based on the location (bucket sort). In the bucket sort, we will be partitioning our list into a number of buckets. Each bucket is then sorted individually. So we will define the number of the buckets based on the number of the salt bodies in my data.

Checking each bucket separately (represents only one salt body where we want to detect the seed), we will define the maximum values in (x,y). and then the difference ΔX and ΔY will be the coordinate for the seed point inside the salt body (See Algorithm 5). Or we can use three points in my list (Bucket) to define the seed.

Defining our seed by using three points from our bucket(sublist) Let (s1,s2) be the coordinates of the center of the circle(the seed) that we want to calculate, and r its radius. Then the equation of the circle is:

$$(x - s1)^2 + (y - s2)^2 = r^2 \tag{4.6}$$

The three points that we will get from our list with high Sobel values will be

Algorithm 5 Automatic detection of the seeds

Require: The input will be the image (The Seismic Data)
  for all $i = 1 \rightarrow ImageLength$ **do**
    for all $j = 1 \rightarrow ImageWidth$ **do**
      for all $k = 1 \rightarrow ImageDepth$ **do**
        Run Noise Removal Filter on (i,j,k)
        Calculate Sobel Value at (i,j,k)
        if Sobel Value $\gtrless$ Limit **then**
          Add it to List
        end if
      end for
    end for
  end for
  for all Points in the List **do**
    Buket Sort based on the number of the salt bodies in my data
  end for
  for all Buckets in the List **do**
    Calculate the MAX, MIN in x,y coordinates
    $Seed = (\Delta x, \Delta y)$
  end for

(x1,y1), (x2,y2), (x3,y3), to estinate the circle inside the Salt Body. Since the three points all lie on the circle (the borders of the salt body), their coordinates will satisfy the Equation 4.6. That gives you three equations:

$$(x1 - s1)^2 + (y1 - s2)^2 = r^2 \tag{4.7}$$

$$(x2 - s1)^2 + (y2 - s2)^2 = r^2 \tag{4.8}$$

$$(x3 - s1)^2 + (y3 - s2)^2 = r^2 \tag{4.9}$$

in the three unknowns s1, s2, and r. To solve these,we will subtract the first from the other two. That will eliminate r, $s1^2$, and $s2^2$ from the last two equations,leaving us with two simultaneous linear equations in the two unknowns s1 and s2. Solve these, and we'll have the coordinates (s1,s2) of the center of the circle (the expected seed).

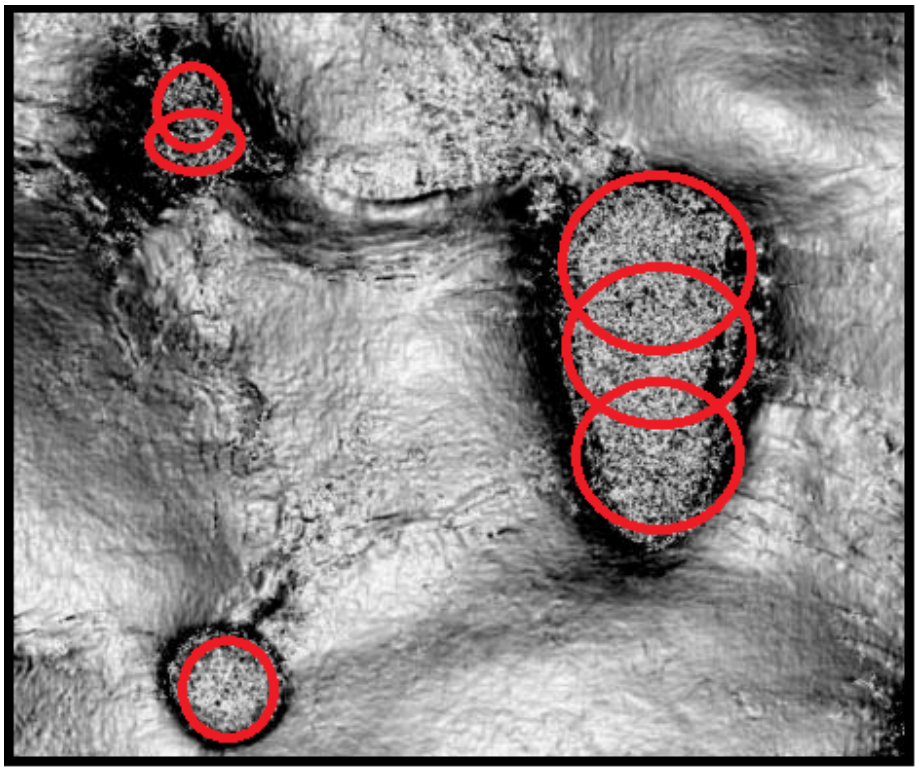
This can all be done symbolically, of course, but we'll get some pretty complicated expressions for s1 and s2. The simplest forms of these involve determinants

(Equations 4.10 nd 4.11):

$$S1 = \frac{\begin{vmatrix} x1^2 + y1^2 & y1 & 1 \\ x2^2 + y2^2 & y2 & 1 \\ x3^2 + y3^2 & y3 & 1 \end{vmatrix}}{2 * \begin{vmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{vmatrix}} \tag{4.10}$$

$$S2 = \frac{\begin{vmatrix} x1 & x1^2 + y1^2 & 1 \\ x2 & x2^2 + y2^2 & 1 \\ x3 & x3^2 + y3^2 & 1 \end{vmatrix}}{2 * \begin{vmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{vmatrix}} \tag{4.11}$$

Figure 4.25 Automatic detection of the seeds



We expect that the salt body will have one or more circle (see Figure 4.25), and these circles can intersect with each other. This will result getting more seeds that will grow together in the same time.(Check Algorithm 4)

Chapter 5

RESULTS AND ANALYSIS

In this chapter, we will follow the same structure of the Methodology and implementation chapter. We will review the results and discuss and evaluate the output, and analyze in the first part of each section, and then focus on the performance that will include the cost of computations, time and memory.

The most important question when we are working on developing our own algorithm is how efficient this algorithm or our implementation (the code that will be implemented) is going to be. Efficiency in this context can cover a lot of resources, including: CPU (time) usage, memory usage, disk usage and network usage. All of them are essential but we will mostly focus on CPU time, and the computation cost. This is due to the algorithm being compute bound.

Since memory usage is a big issue in seismic data, and a lot of efforts were given by other research in their publications focusing on the compression of data and memory challenge, we will skip this part in our analysis, and keep our focus on computation analysis.

In our discussion, we will differentiate between two terms:

- Performance: This is directly related to the program running, and then we can detect exactly the used time, memory or disk. This depends on the machine, compiler and the code. As we mentioned before that the algorithm has been implemented in Matlab in our work. So the information related to the performance will be considered as indicators in the comparison cases but not consider as exact information. In addition, the tests were done on personal laptop while in reality they will be run on better quality machines.
- Complexity: This will cover how the resource requirements of a program or algorithm scale, considering the different cases when the size of the program being solved gets larger.

As a rule, complexity affects performance but not the other way around. So we will focus on the complexity of the algorithm and then show tables representing the performance parameters in our tests.

The time required by a method or algorithm is proportional to the number of the basic operations that we have. the basic operations includes: the arithmetic operation, assignment values, test and condition as if(X==Y), read operation or write operation. In general, we can classify the algorithms into two types: algorithms can perform the same number of operations every time they are called that will require constant time always. The other type of the algorithms that perform different numbers of operations, depending on the value of a parameter or a field. We call the main factors and the parameters whose values affect the number of operations performed the problem size or the input size. In our work, we used the second type. All the different algorithms that are using are mainly depending on the input data size, and then the list that will be created to define the pixels to be defined as edges or not. We will explain in more details each section while we are evaluating the complexity.

To calculate the complexity of our work, we will look to the relation between the number of operations relates to the problem size. Not to the exact number of operations that are performed. We will consider the worst case, which is the total number of operations that might be performed for a given problem size.

5.1 Seed growing with Sobel Values

In this section , we will represent the results of running seed growing algorithm based on the normalized Sobel Value (See Section 4.3, Figure 4.11 and Algorithm 1).

5.1.1 In terms of output

Looking to Figure 5.1, we can see that the algorithm is working as we expected. It is starting from the seed (the red dot in the circle to the left) and growing slowly by checking the Sobel value and adding the pixels that satisfies the condition to the list.

In Figure 5.1, the component are black and white, there is no noise to disturb the growing manner. So this is the basic case that we will build on during our work.

We tested our algorithm on seismic data, see Figure 5.2, the border of the salt were detected, however the noise effected our results. We can see that in the marked Blue area (that we predefined in our implementation), we still have some black dots inside the salt body (the black dots inside) which represent ”edges” in our implementation.

Figure 5.1 Results: Sobel-Seed Growing

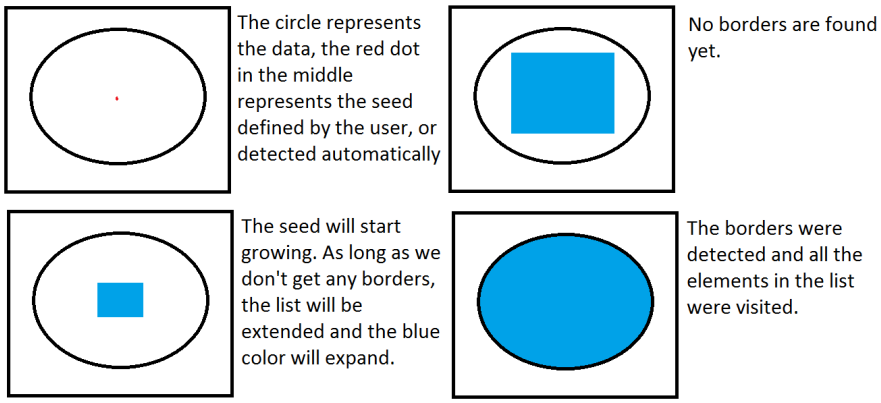
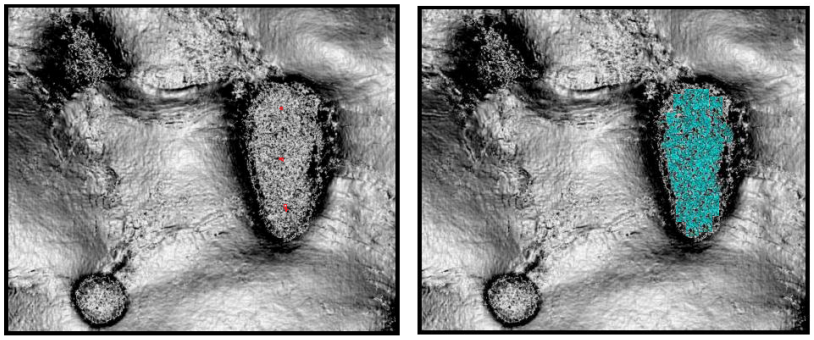


Figure 5.2 Result: Sobel-Seed Growing on seismic data



We will start from the red dots (The seeds), our Sobel limit is 0.2

Results of Running Seed growing-Sobel Filter

This can be explained as follows, in the noise pixels, there are differences in the intensity because of the noise. So the Sobel values in these pixels are going to be larger than the limit we defined in our condition, which will result in defining it as an edge and color it black. In the next step, we will show how we solved the noise issue in our growing algorithm.

5.1.2 In terms of performance

The seismic filtering algorithms are considered to be very data intensive, so we will discuss the computations cost, the used time and the memory constraints. Taking in consideration, that seismic processing tasks are usually performed using machines with high speed big storage devices. While our tests and experiments were run by average personal laptop.

The complexity of the seed-growing-Sobel algorithm

If we assume that the maximum number of pixels that we want to extract is going to be N . Then the complexity of the algorithm is going to be calculated as follows:

- The first step to extend 8 points around the defined pixel: considers as assigning values to the list, is a basic operation will have constant cost.
- The while loop: will be checking if all the points in the list are visited (the list is growing inside the loop continuously based on the termination condition) will run at least $log(N)$ times.
- The for loop inside will run from 1 up to the size of the list (N), inside the for loop we have :
    1. Calculating Sobel Value: basic operation which have a constant cost.
    2. If statement: to check if Sobel value is greater or less than the Limit, and this is a basic operation as well with constant cost.
    3. the inner loop to extend 8 more points, and to check if the new added pixels are exist already in my List or not: is going to run 8 times every time. It will not be affected by the number of the visited pixels (the size of the target body) so it will consider as a constant cost as well.

As a result (Figure 5.3): we can predict that our algorithm will have the following complexity: $(N * log(N))$.

In Figure 5.4, we used the clock in Matlab (Tic-Toc function) to measure the performance of our algorithm running on Seismic sample (See Figure5.2). With 24576 bytes as size, represented in Matlab in an array 106 *145*3. The x-axis represents the number of the visited pixels in our growing algorithm, and the y-axis represents the time. We can see that it takes the shape of the logarithmic function.

Figure 5.3 Calculating the complexity for seed-growing-sobel algorithim

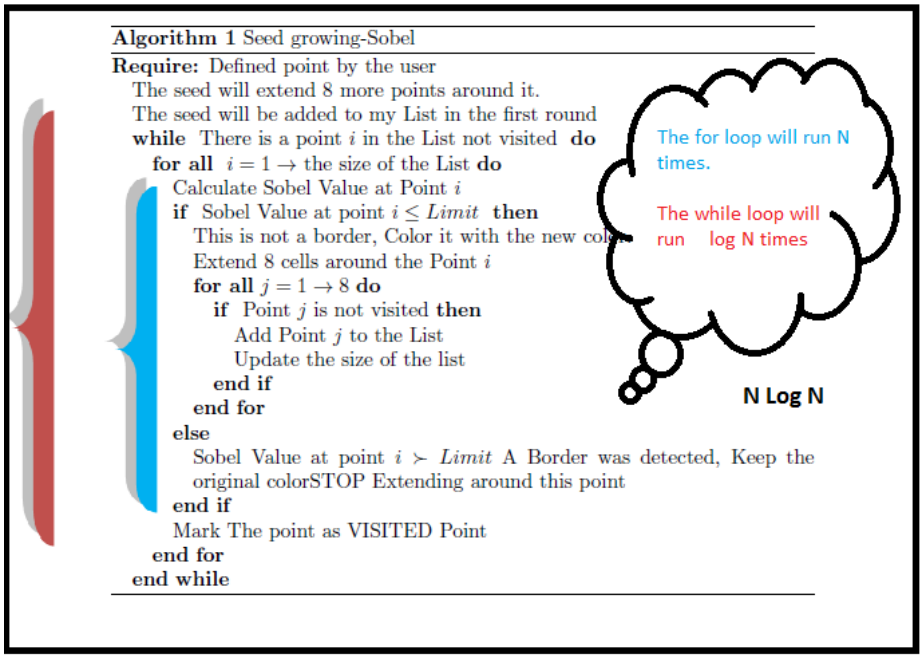


Figure 5.4 Tic-Toc trace for seed-growing-Sobel algorithm

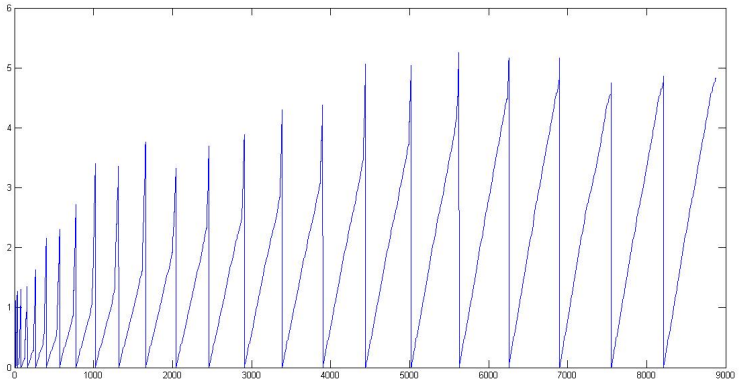
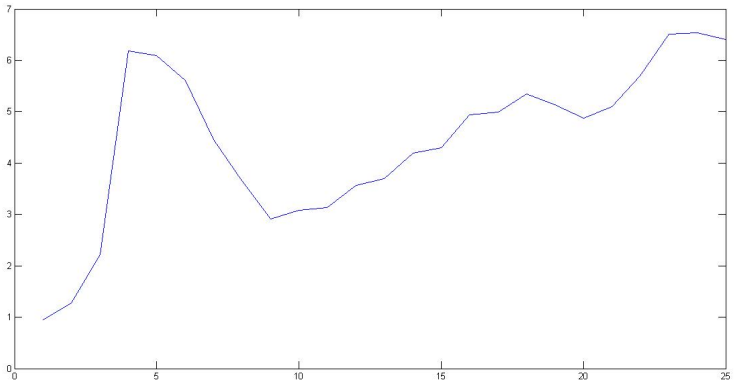


Figure 5.5 Tic -Toc trace for seed-growing-Sobel algorithm



When we started with the seed, the list was almost empty, so the check condition in the while loop was faster to be checked, and the possibility to extend was easier. Later on, the added pixels to my list after the extension make the condition in the while loop goes slower (since the list is growing like a tree with the help from all the children)and the possibility to extend is less, since we have many pixels already in the list. Later on, it goes faster, because the edges were detected and the extensions were stopped in some parts(leaves) of the list(tree).

In Figure 5.5, we measured the performance when extending a new cluster (when a new extension process is going on). In this figure, we can notice easily the effect on the growing level when detecting an edge, the growing level is going down. One leave will stop and the rest will continue growing until the rest of the edges are detected. This shows our motivation to use the seed growing algorithm. We will extend and process the data with different filters ONLY when it is needed. Instead of processing the whole image which can cause blurring or affecting the small details in our data. In addition to the fact that the required storage and time is going to be less.

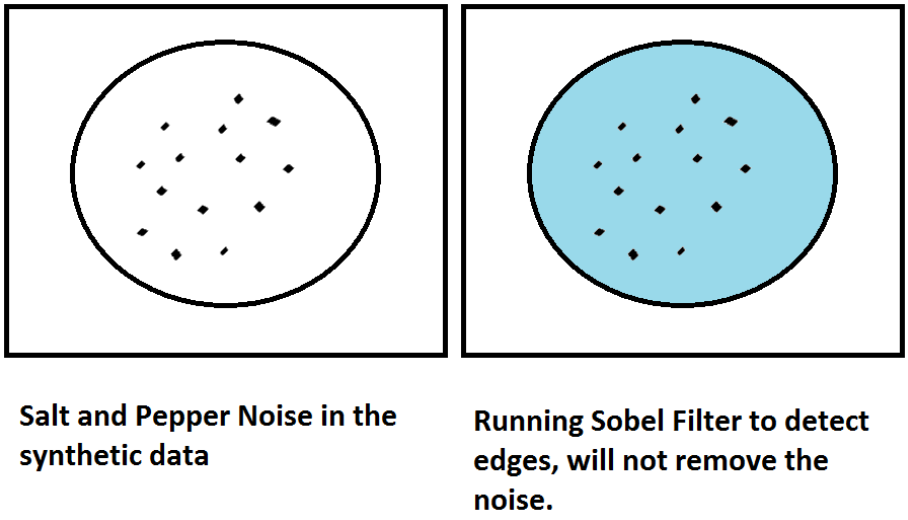
5.2 Seed growing with noise

In this section, we will represent the results of running seed growing algorithm combined with the noise removal filters (See section 4.4, Figure 4.14 and Algorithm 2).

5.2.1 In terms of output

We will start with Figure 5.6. This figure shows the results of running Sobel Filter on our synthetic data with noise. The black points in the middle of the circle (the noise) were detected by Sobel Filter as edges. This is what we expected because there is no smoothing filters there, and the values in the noise pixels are high and as a result the Sobel value is high and comparing it to our condition, it will look like 'edge'.

Figure 5.6 Running Sobel Filter on the data with noise.



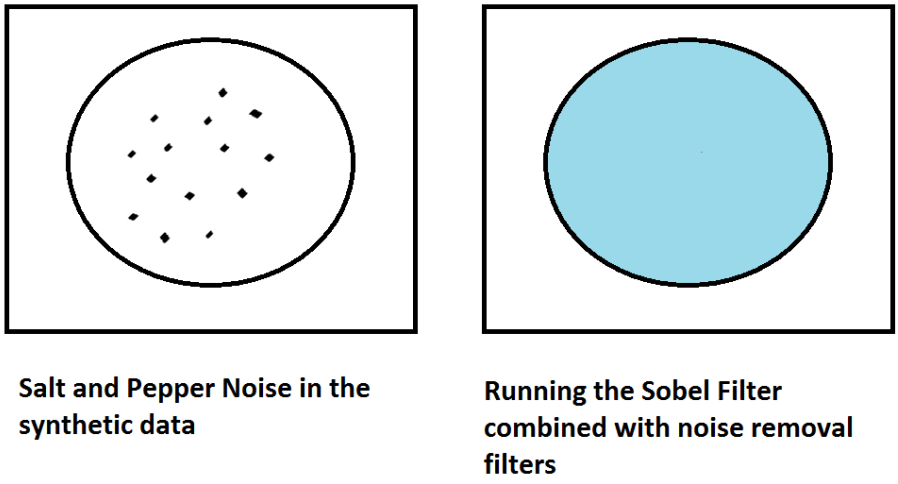
The use of the noise removal filters is to check the surrounding pixels to check if these high values represent noise or real edge. Starting with the synthetic data, we tried to run the noise removal filters (median, adaptive median, mean and Gaussian).

See Figure 5.7, the results of running the median filter before Sobel in seed growing algorithm. We can see that the noise was removed and the real edges were only detected. Running the median filter with 5*5 scale, allow us to compare the noisy pixels with their surrounding pixel and then getting the median value to replace, eliminate the noise and keep the edges.

Running the adaptive median, mean and Gaussian filters on our synthetic data will give us the same exact results as Figure 5.7. The reason is that our data is black and white and we will not be able to see the differences between these different filters.

With the median filter (and the adaptive median), we have high contrast values that are scattered. Then it will basically output white color, because the median

Figure 5.7 Running Sobel-Noise Removal filters



will be white as long as there is more of it in the sample size (the scale of the surrounding neighbors).

In the mean case (and Gaussian), then the result for the filtered area is gray since that is the average of white and black values on the area. However it will still work, because of the threshold value we defined on Sobel, and because the difference between (gray and gray AND white and gray) in contrast is small and will give us a small Sobel value. That will color it in the predefined color that we can see in Figure 5.7. To compare between the different noise removal filters, we will test the different filters using sample from processed seismic data (See Figures 5.8, 5.9).

5.2.1.1 Median and adaptive median filters

In this section, we will compare between the median and the adaptive median. Because they have the same concept to filter with some addition conditions in the adaptive Median. It is interesting to see the effect on seismic data.

Figure 5.8 shows the results of running the median (to the left) and the adaptive median (to the right) filters combined with Seed growing algorithm based on Sobel values. We can see that the noise was eliminated and the borders of the salt bodies were defined clearly. The performance of the adaptive median in the noisy ambiguous borders was better (the last case in the figure the top left).

The filters remove the noise and kept the fine details in our sample as well. In the adaptive median the filtering was performed only in the noisy pixels in a way that the details in the other pixels will not be effected. We can see that we

do not see blurring in our results, since we are not running the filters for the whole image. Just in the pixels that we will extend.

5.2.1.2 Mean and gaussian filters

In this section, we will compare between the mean and the Gaussian filters. Since they have the same concept as well to filter following the Gaussian equations in the last case. We will observe the effect to our data.

Figure 5.9 shows the results of running the mean (to the left) and the Gaussian (to the right) filters combined with Seed growing algorithm based on Sobel values. We can see that the noise was eliminated and the borders of the salt bodies were defined clearly. The performance of the mean in the noisy ambiguous borders was better. This can be noticed in the small noisy bodies (left bottom, left top and right bottom in the same figure).

The mean filter combined with Sobel gave us results better that Gaussian filters. The extension for new pixels were less in the last case. because filtering using Gaussian and then running Sobel gave us high Sobel values which stopped the extension in our case and defined the noise as edges and stop before reaching the real borders of the salt bodies.

5.2.1.3 Mean and median filters

In this section, we will compare between the median and the mean filters and see what makes one of them is better than the other on the seismic data.

we can see in Figure 5.10 that the mean filter works better in the first two cases, while the median filter works better in the last two cases. This can be explained that the median filter remove the noise while the mean filter spreads it around.

In the case of the impulse noise, the median filter will work better, but it can remove some small details and this important in our data. However, as we explained in the previous section the adaptive median filter can give us better results with seismic data. Since the median value must actually be the value of one of the pixels in the neighborhood, the median filter does not create new unrealistic pixel values when the filter straddles an edge. For this reason the median filter is much better at preserving sharp edges than the mean filter. The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly

while the mean filter can work better when the noise is surrounded by fine pixels, and it will keep all the details.

Figure 5.8 Comparing the results from median(left) and adaptive median(right) combined with seed growing based on Sobel Filter

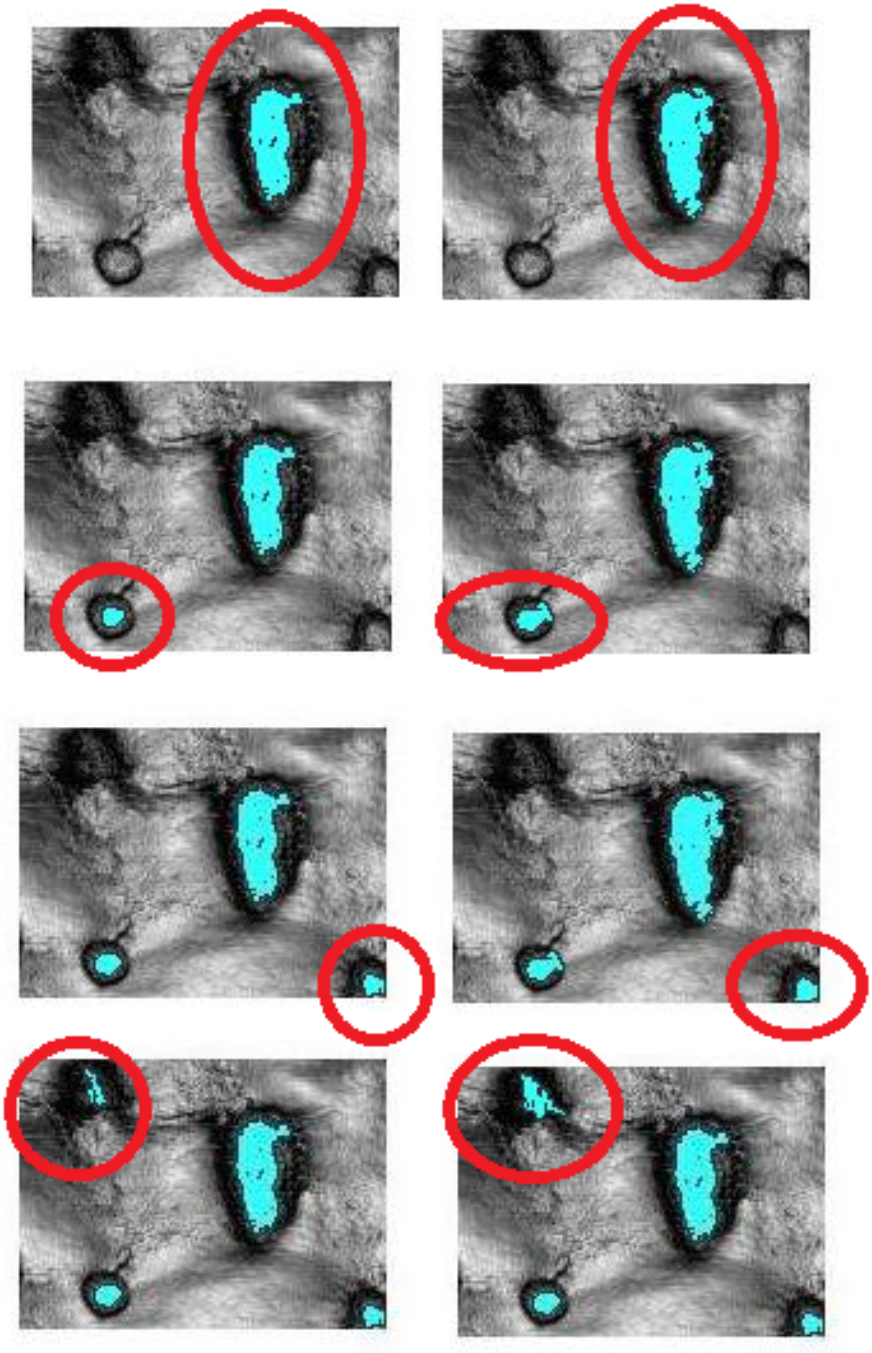


Figure 5.9 Comparing the results from mean(to the left) and Gaussian(to the right)combined with seed growing based on Sobel Filter

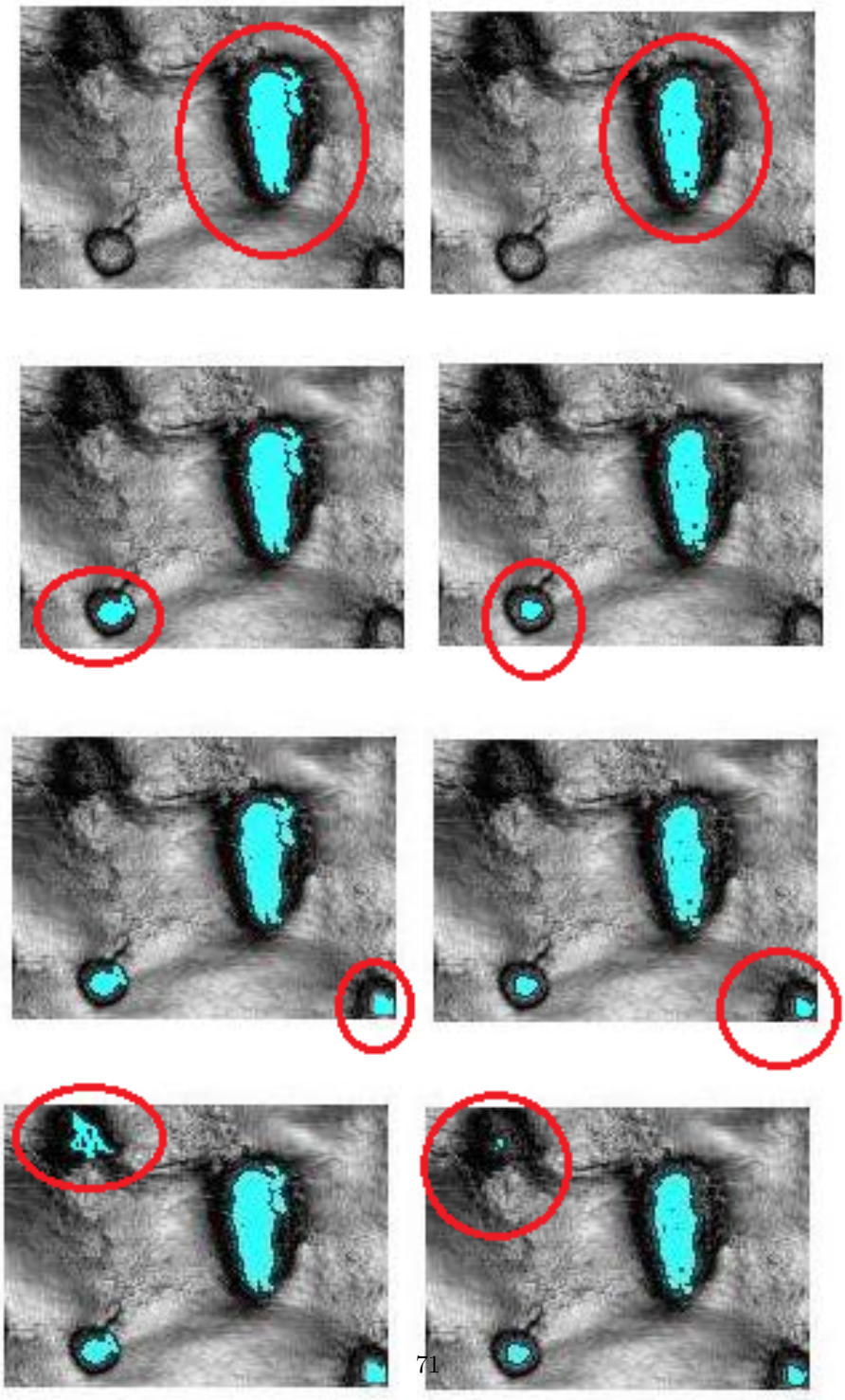
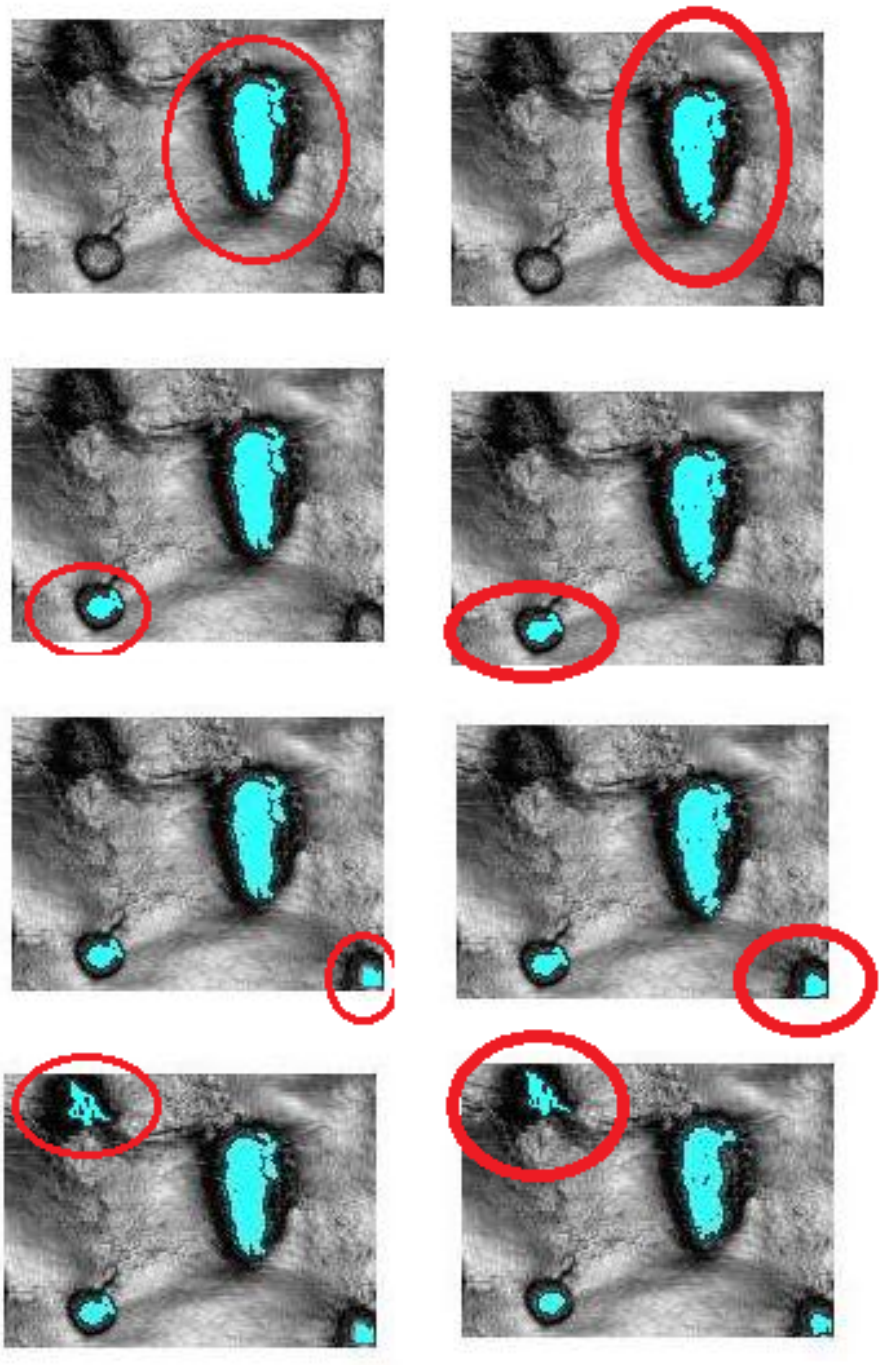


Figure 5.10 Comparing the results from Median and Mean filters combined with seed growing based on Sobel Filter



5.2.1.4 Comparing the use of different sizes for the filter window

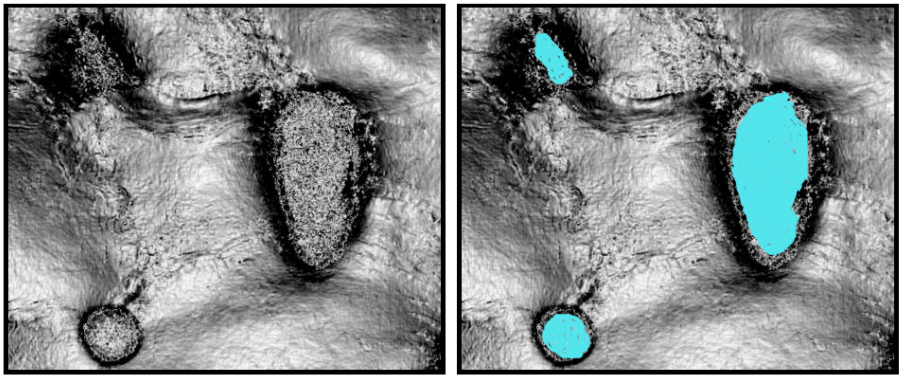
We compared using different sizes for the filter window. The filter window is the nearby neighbors of the targeted pixel, to decide whether or not it is representative of its surroundings. We had tried 3*3 square neighborhood and 5*5 and 7*7. The larger neighborhoods will produce more severe smoothing.

When we smooth the noisy image with a larger filter, e.g. 7*7, all the noisy pixels disappear. However, the image is beginning to look a bit blotchy, as gray level regions are mapped together. Alternatively, passing a 3*3 median filter over the image three times in order to remove all the noise with less loss of details, gives better results.

5.2.1.5 Hybrid combination of smoothing methods

After we analyzed the output of running different filters and their effects on the seismic data, we recognized that each filter will work in optimal way in some cases. These cases depend on the noise nature in the seismic data and the values of the pixels in the same neighborhood. So we run the hybrid combination of smoothing filters (Section 4.4.5). The switching between the different filters based on the value of the variance of the neighborhood in the same run, gives us the best result (See Figure 5.11).

Figure 5.11 Hybrid combination of smoothing filters

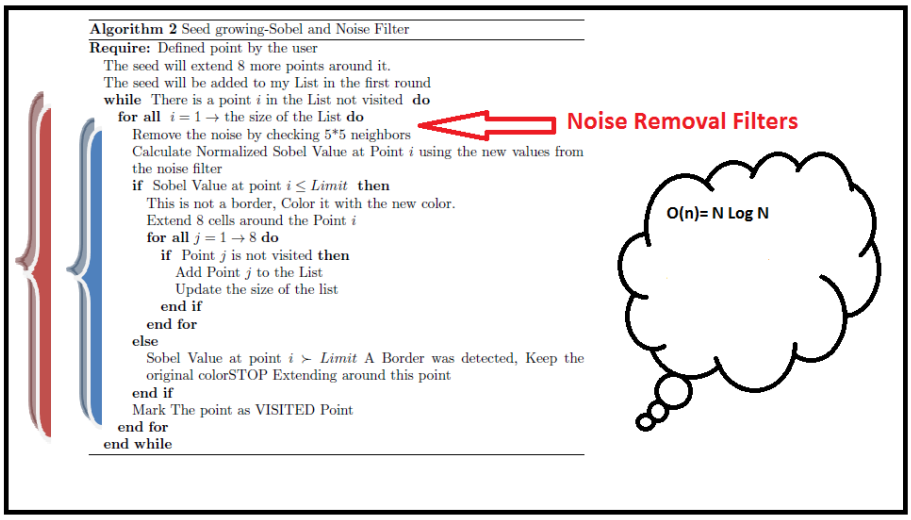


5.2.2 In terms of Performance

To calculate the complexity for seed-growing-Sobel algorithm combined with the noise removal filters (See Figure 5.12). As we can see the addition to our

original algorithm is going to be running the smoothing filters before running Sobel filter.

Figure 5.12 Calculating the Complexity for Seed-Growing-Sobel Algorithm combined with the noise removal filters



Our filters (median, adaptive median filters, mean and Gaussian) calculations are going to be linear. The main common two steps are:

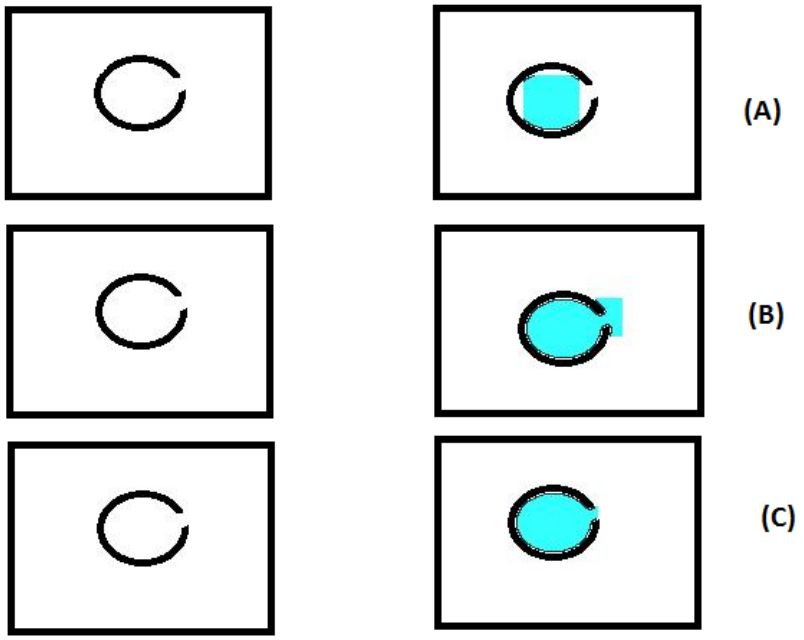
1. Extend enough pixels for my scale (5*5, 7*7,9*9): which is assigning values and add it to my list which is basic operation, that has constant cost.
2. Do the calculations on the defined list:
   - Finding the median in the list: order the list and then take the median value to replace it with the processed pixel.
   - Finding the adaptive median in the list : classify the noisy pixels and replace the median ONLY there, and leave the fine details, More conditions were added here. However, the total cost still less than linear.
   - Finding the mean in the list : the mean tool considers as linear tool. After defining our list, we will find the summation of the value and divide into the number of the elements in the list.
   - Finding Gaussian value: by doing the basic calculation for Gaussian. At the end, the complexity for my algorithm after adding the removal noise part is going to stay the same $(N * log(N))$.

5.3 Disconnected areas

In this section, we will represent the results of running seed growing algorithm combined with the noise removal filters and the boundaries estimation (See Section 4.5 and Algorithm 3).

5.3.1 In terms of output

Figure 5.13 Results of running different termination (directional history)



In this section, we will represent the results of the directional history approach that we implemented. In addittion to the different termination conditions that we tested on both our synthetic data and processsed seismic data.

In Figure 5.13, we run our connectivity function with different conditions to observe the behavior of the growing. In the first Figure A, we calculated the directions in comparing to the original seed, the termination condition was to stop if the growing in one direction was higher than the 1 of the total size of the list. This is because we are growing in 8 directions in the same time. As we can see the growing stopped earlier than it is supposed to.

Then we modified our condition in the same Figure B, to check the whole area not only the direction (we have 8 directions divided into 4 areas). And the

results were better by covering all the body, however the growing this time was more than what we expected, the combination of the various stages of our algorithm are not affected by the edge estimation.

So a new modification was required for results in Figure 5.13.C . A combination between the directions and the areas was implemented here. We will check first the balance between all the 8 directions and then if unbalanced behavior was detected, we will not stop directly (it will be so early as we see in A), we will check a second condition related to the balance of the growing in the different areas and then if that shows unbalanced behavior we will stop, otherwise we will continue. In part C in the same figure, we can see that the results were very good. The estimation to the missing edges was totally percise and logical. In

Figure 5.14 Steps Showing our growing algorithm with bigger discontinuous area

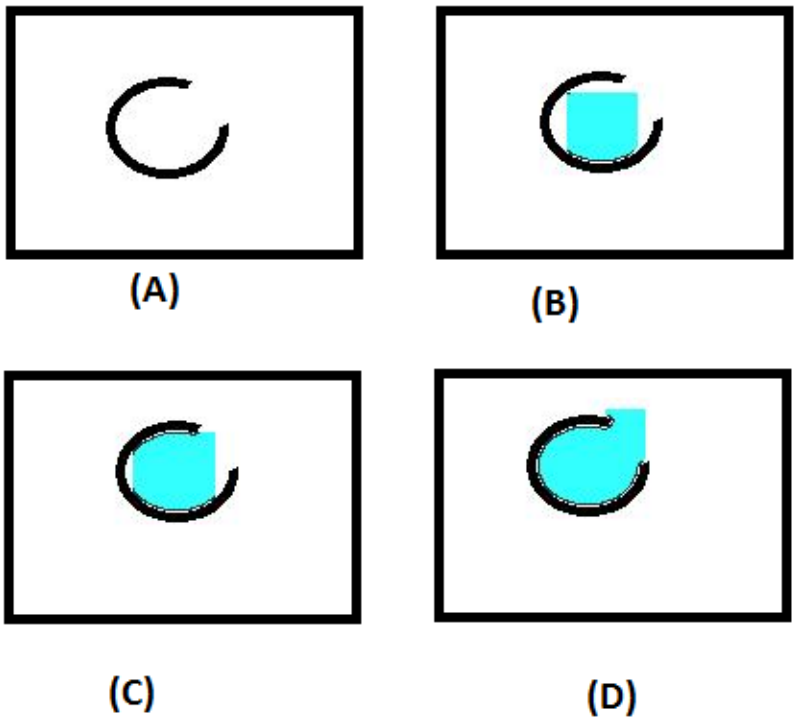
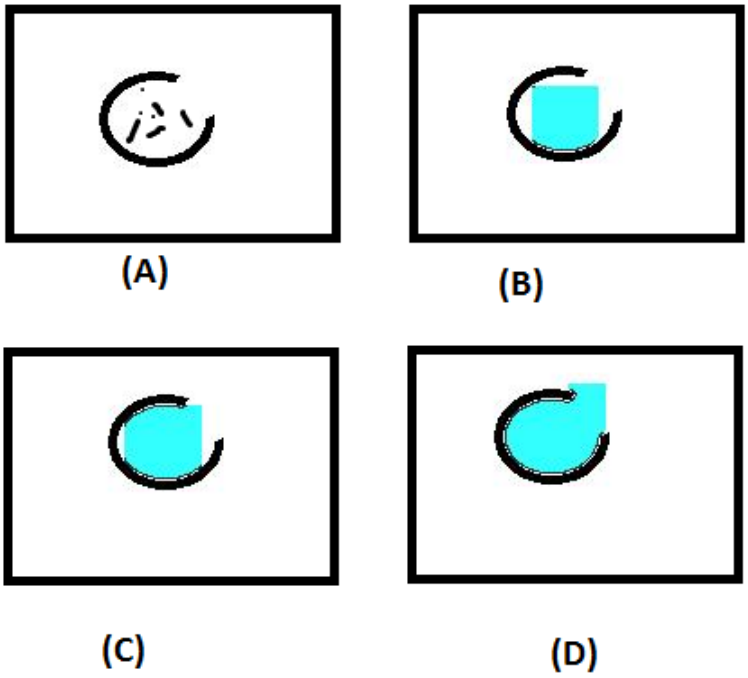


Figure 5.14, we tried to run the algorithm with bigger discontinuous area (Part A in the Figure). The steps show how the growing in our case is taking square shape (the 8 directions growing) in Part B in the Figure. When detecting some borders, the growing is going to be extended in the other directions (right and left in our case) in Part C. At the end the edges were detected in the left side

and the open discontinuous side will grow a little bit outside the body and then it will be stopped. The bigger the discontinuous areas become, the less accurate estimation will result. In Figure 5.15, we added the noise to our sample. The

Figure 5.15 Steps Showing our growing algorithm with noise



results are as we expected. Now we verified that the combination between our seed growing, edge detection, noise removal and the estimation the disconnected areas are working together. In Figure 5.16, we can see the results on seismic data. The noise was removed, the growing stopped after reaching the borders and the estimation for the disconnected areas was so close to the shape of the body. The extraction of the body in the new color can easily help the observer to see the salt body completely.

5.3.2 In terms of Performance

To calculate the complexity for seed-Growing Algorithm combined with the noise removal filters and the directional history to estimate the missing borders of the salt, we will look to Figure 5.17.

In Figure 5.17, we can see that the addition in this case, was the check for the

Figure 5.16 The results of Directional History on Seismic Data

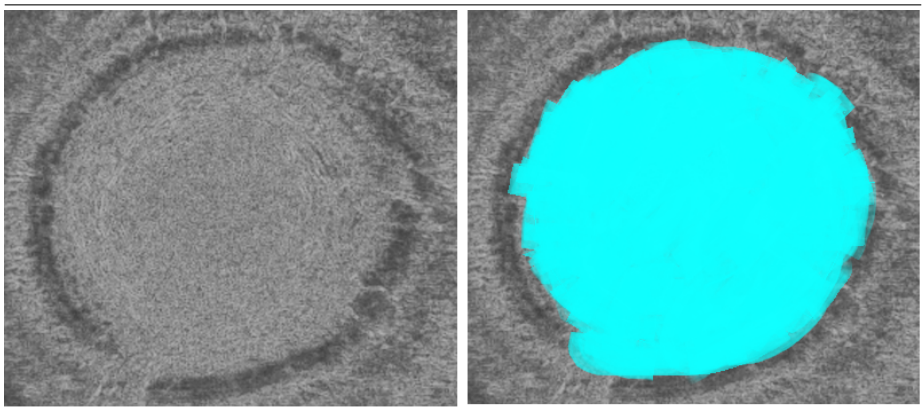
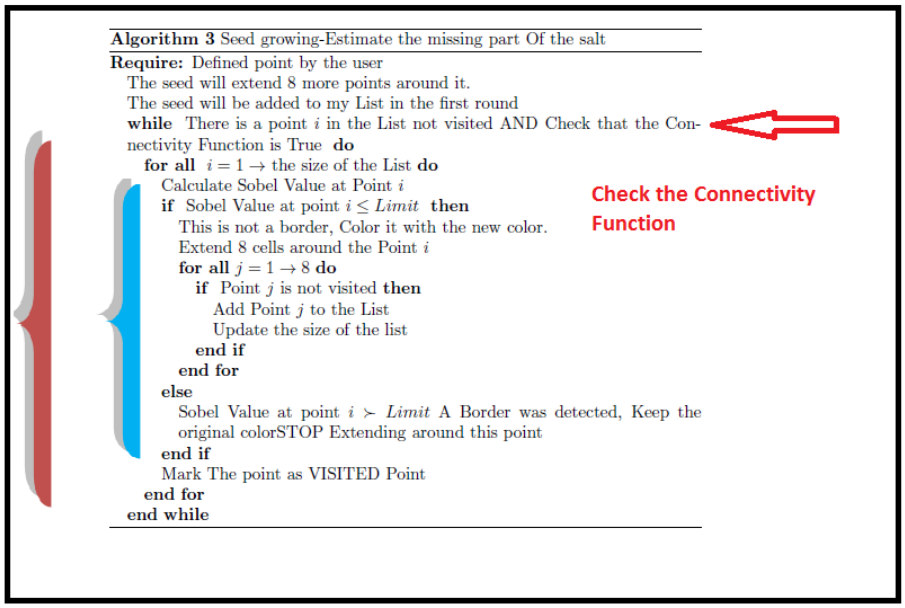


Figure 5.17 Calculating the complexity for Seed-Growing Algorithm combined with the noise removal filters and the directional history to estimate the missing borders of the Salt bodies



connectivity function. To see the additional cost caused by this addition, we will explain briefly the connectivity function.

When our seed is growing for the first time, 8 counters will be initialized representing the different directions. Every time we will keep growing in the same direction,+1 value will be added to the counter, while -1 will be added if we are growing in the opposite direction. All that will not be costly since it is assigning values and consider as Basic Operations.

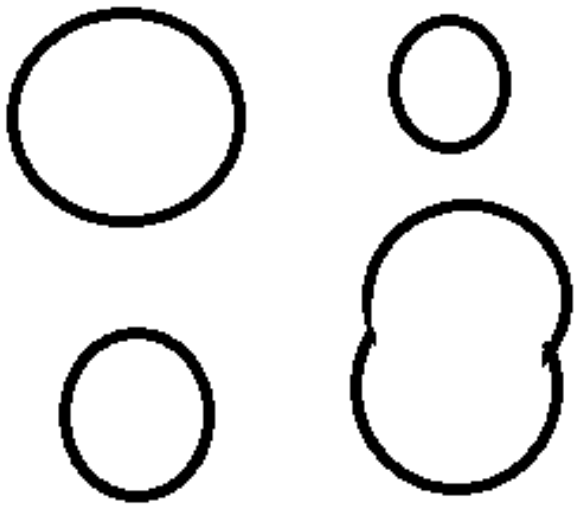
In the check condition function, we will have many different conditions in the form of IF-ELSE. This is a basic operation as well. All the cost added by the directional history will be considered as a constant cost. The Algorithm will keep $(N * log(N))$ as cost as we explained earlier in the first case.

5.4 Automatic detection of the seeds

In this section, we will present the results related to the automatic detection of the seeds. As we suggested in section 4.6.1, algorithm 4, we will check every pixel in our image and check if that pixel is a seed (giving the minimum and maximum values of the radius, See Figure 4.24). Furthermore, we will represent the results of the second approach (section 4.6.2, algorithm 5); where we detect our edges and consider them a circle or combination of intersected circles and the seeds will be the centers of the circles.

5.4.1 Check if the targeted pixel is a seed point

Figure 5.18 Synthetic input data (automatic detection of the seeds)



In figure 5.18, we show our synthetic data to evaluate our approaches to detect the seeds. We suppose that the salt bodies are circles or combined intersected circles (More information about this assumption see section4.2.1). In figure 5.19 and table 5.1, we can see the results on synthetic data when we run Algorithm 4. We will calculate the sum of the surrounding pixels in circular shape around each pixel. If the value is high, then we define this pixel as a seed and it will be colored . In figure 5.20 and table 5.2, we can see the results on seismic data(we are using 5x5 grid to visualize the seed), the detection and seeding are all within the salt bodies such that we are only segmenting them. This is important to avoid growing outside of the wanted structures.

Figure 5.19 Results on synthetic data: Algorithm Check if the targeted pixel is a seed (Algorithm 4)

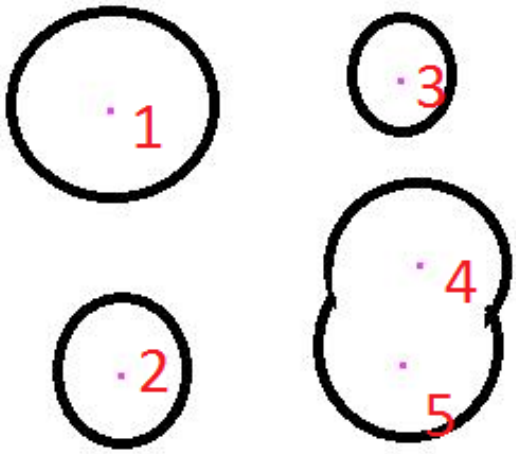


Figure 5.20 Results on seismic data: Algorithm Check if the targeted pixel is a seed (Algorithm 4)

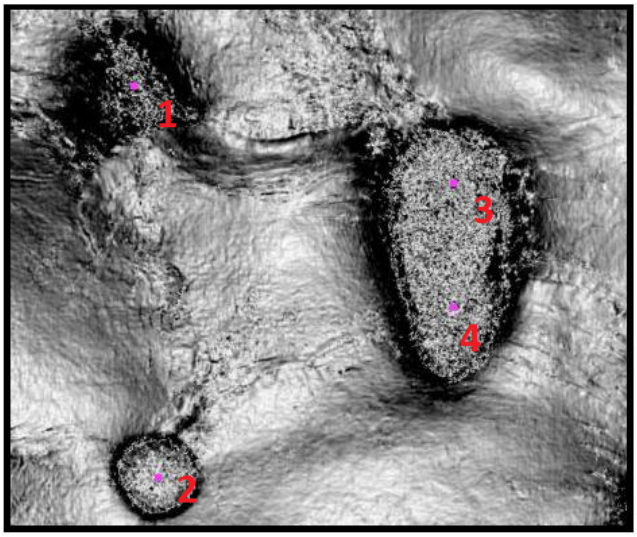


Figure5.19	X-coordinate	Y-coordinate
Seed 1	79	214
Seed 2	79	146
Seed 3	205	222
Seed 4	82	205
Seed 5	63	214

Table 5.1: This table shows the detected seeds coordinates

Figure 5.20	X-coordinate	Y-coordinate
Seed 1	64	99
Seed 2	350	115
Seed 3	144	333
Seed 4	230	337

Table 5.2: This table shows the detected seeds coordinates

5.4.2 Define my edges and then calculate the seed based on

In this section, we will test the second approach. We will detect the edge pixels, and then sort them based on the location, after that we can define the center of each body as the difference between the minimum and maximum in the coordinates. In Figure 5.21 , we can see that the first step is to define the borders and then the seeds are the centers in the detected bodies (Figure 5.22 and table 5.3).

Figure 5.22	X-coordinate	Y-coordinate
Seed 1	108	208
Seed 2	160	301
Seed 3	235	96
Seed 4	165	296

Table 5.3: This table shows the detected seeds coordinates

Figure 5.23 and table 5.4 , show the results of detecting the seeds on seismic data. In general, we can conclude that both approaches give the required results by detecting the seeds within the salt bodies. The techniques are different in the method to detect the seeds. When we have a salt body that is combined of two intersected circles, the first approach will detect two seeds in the middle of the body, this can be effected with the maximum radius that the user predefine. However, in the second approach we are detecting the edges and sort it based on the location. We define them in a list and the list will be sorted and the differences will define the center of the bodies. This is the reason that we got one seed in the big salt body in comparing to 2 seeds (in the first approach).

Figure 5.21 Results on synthetic input data: defining the edges in the first step in Algorithm 5

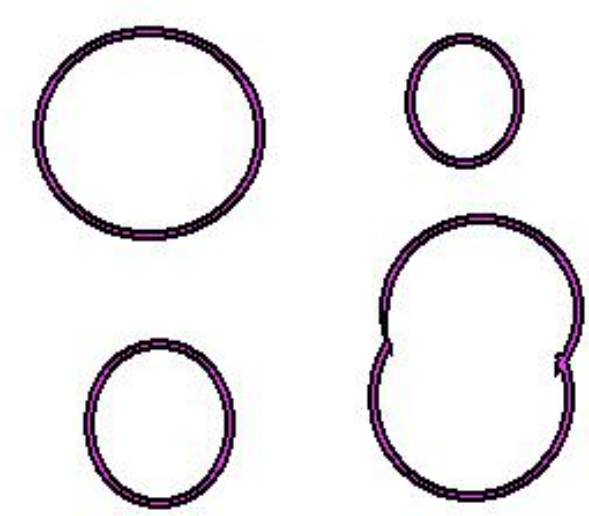
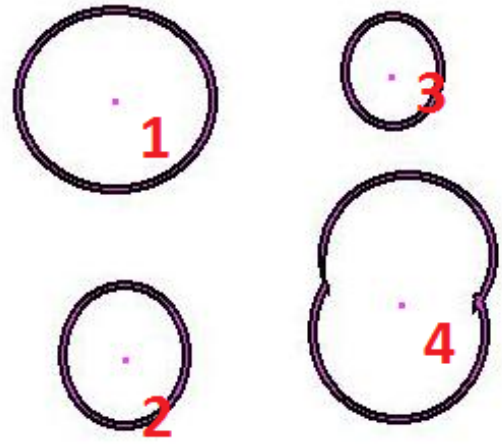


Figure 5.22 Results on synthetic input data: defining the seeds in the second step in Algorithm 5



The quality of the two approaches will depend on the input data, if we have data with ambiguous borders, the first approach will give better results. This is because we are checking every pixel and the surrounding pixels. However, the second approach can be very accurate if we have fine data with sharp borders. Combining both approaches and comparing results, can give us better performance and precise detection of the starting seeds in our growing algorithm.

Figure 5.23 Results on seismic data: by defining the edges and then the seeds using Algorithm 5

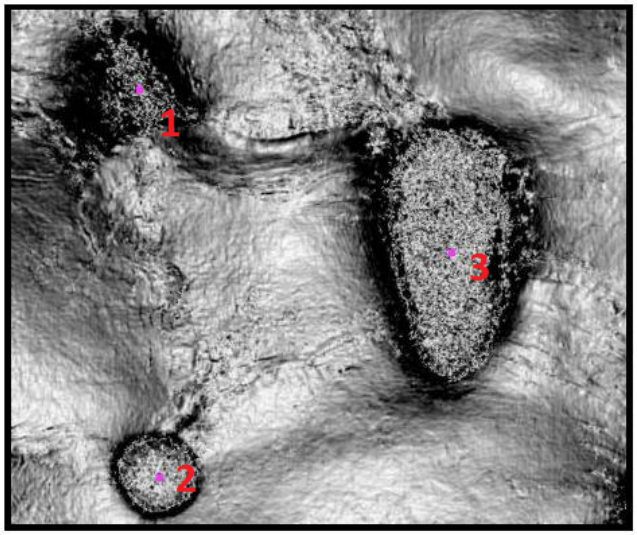


Figure 5.23	X-coordinate	Y-coordinate
Seed 1	63	115
Seed 2	99	184
Seed 3	352	333

Table 5.4: This table shows the detected seeds coordinates

Chapter 6

CONCLUSION AND FUTURE WORK

Exploring for hydrocarbons below salt has been, and will continue to be, a high risk/high potential business endeavor. Recent subsalts discoveries have generated a tremendous interest in subsalt exploration, both in the Gulf of Mexico as well as in other basins around the world. The salt and raft tectonics in the Basin is considered as one of the main factors that influenced the distribution of oil reserves in many regions in the world, and its understanding is very important for hydrocarbon exploration. Proper stratigraphic interpretation of a seismic dataset is a huge task requiring months of effort. Still, the seismic contractors around the world introduce more and more acquisition capacity. Hence, if we want to extend seismic exploration to enable detection of these potentially vast reserves, we need tools capable of handling the new trap models.

There exist several approaches where an image filtering algorithm is used to detect such structures; such is the case with well known seismic attributes like variance, coherence, amplitude contrast, etc. These attributes usually are a good indication of the flanks of the salt structure, but struggle in segmenting the salt body as a whole.

In our work in this theses, we suggest a novel method of seeded growing to aid in this task. The novel aspects are that we account for noise reduction/removal, salt boundary detection, and incomplete boundary estimation and termination. As input to our algorithm we consider filtered seismic data with typical edge detection seismic attributes.

The next section, Section 6.1, summarizes our results, including the output of implementing our method in Matlab, the algorithm performance ,and our results impact on the seismic dataset. Possible future work within the field is presented in Section 6.2

6.1 Conclusion

The approach of a seeded growing algorithm to segment salt bodies proves to be useful, and given a proper filtered input volume and the fine tuning of parameters is able to isolate the salt body as a whole. The approach presented in our work, with the modifications, is effective at this task only when combined with methods of noise removal/reduction and boundary termination estimation for existing and discontinuous boundaries.

Using the edge volumes as input we get a good indication of the flanks and borders of our salt structures. We chose a seeded growing approach for the segmentation because this way we can rely the positioning to be within the salt body. Once a point is identified within the salt body, we can isolate the salt structure. We suggest an automated approach to identify seeding points because it results in a more global solution. While investigating this, we identified two methods. One is where we iterate over all the pixels in the image and recognize if it is a potential seed. The other, where we approximate circular shapes from the salt structures and define the seeds in the centre. Both methods prove to be useful if the input volume is of sufficient quality. In other cases, we can rely on the user to identify the seeding points.

As we are growing within a salt body, there will be the challenge of identifying and distinguishing boundaries from noise. Given the noisy texture of salt data in seismic, we decided to precondition our growing seeds and their neighboring samples with the use of smoothing filters (median, mean, Gaussian, adaptive median). These different filters result diverse quality in the output depending on the dataset that we are testing. So we have also taken into account when to use the various filters for best results and suggest to switch between them give an analysis of the how chaotic the considered neighborhood is. This means that we are using a hybrid combination of smoothing algorithms.

As the noise is reduced or filtered away, we then run a normalized Sobel algorithm for edge detection. This is to detect if we are near any salt boundaries and to terminate our growing in this direction. One is not limited to using the Sobel algorithm here, but can also use others in the coherence/semblance family of attributes for edge detection.

Finally, another unfortunate characteristic in salt data is that it may have discontinuous edges and hence a growing algorithm will grow beyond the salt structure. Our attempt to solve this involves estimating the boundary along discontinuous parts of the structure and terminate accordingly. There are several approaches that could be used for the termination estimation, such as using the double derivative and edge enhancement techniques. However, we have focused on evaluating the growing directions of our seed points and terminate based on unbalanced behavior.

When it comes to the performance complexity of our algorithm, which is $O(NlogN)$.

we conclude that combining the different stages (seed growing, noise remove, edge detection and borders estimation) does not effect the complexity.

6.2 Future Work

Presented below are suggestion for future work within the researched field:

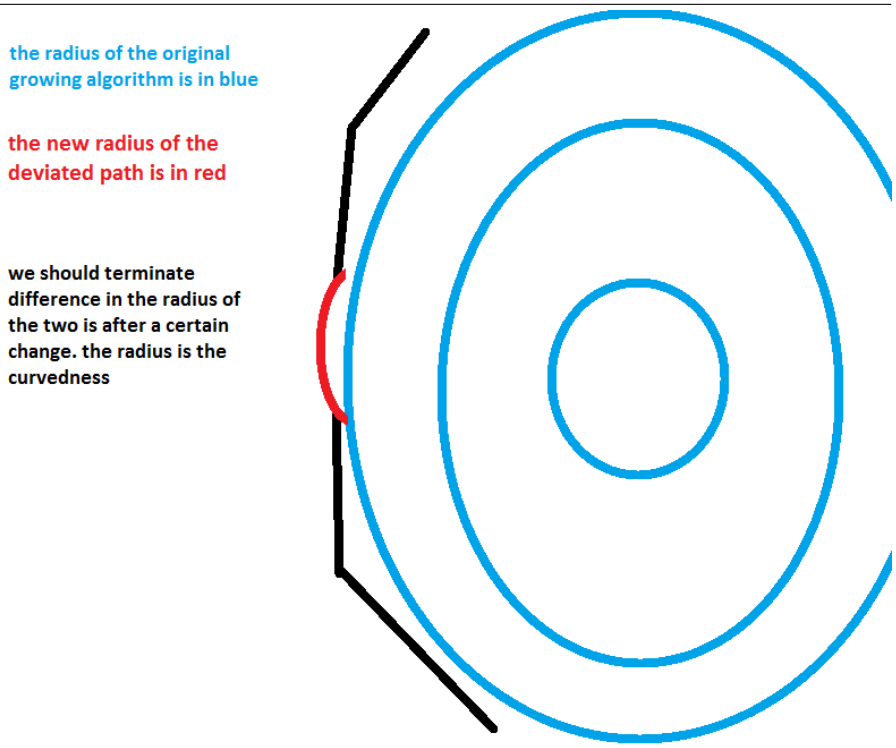
Implementing Our Algorithm in Ocean: Ocean is an application development framework with the capability to work across data domains. It provides services, components, and a common graphical user interface that enables efficient integration between applications. It allows application developers to interact with Ocean model-centric applications like Petrel. Ocean applications are loaded dynamically as .NET assemblies. These assemblies, the building blocks of Ocean, contain modules. So we suggest that our implementation in Matlab can be converted to .Net (Ocean Framework). Our Input Data is preprocessed in Petrel. So adding the whole algorithm as a workflow in Petrel will allow us to start from raw Seismic Data where we expect to have the salt domes and then run the Automated Seeded Growing Method to delineate the salt bodies. In addition to, a comparison between the performance of our algorithm (between Matlab, and Ocean) will be interesting. Taking in consideration, the differences between .Net as object oriented language and Matlab as high level language well known with the matrix computations, numerical analysis and graphics viewing.

Curvature as a termination condition: One of main challenges to detect the salt domes is the discontinuous nature of their boundaries and how to terminate to avoid growing into other structures. We used the historical direction in our growing algorithm , another approach can be useful and interesting is to have a check condition on the curvature while our seeds are growing.

In Figure 6.1, we can see the original growing in our algorithm is represented by the blue color. We will start from the seed and grow around it in all directions and extend further. If we reach the discontinuous areas in the salt bodies, then it will start to take the red shape by deviating the original path. A termination condition can be built on the relationship between the blue curves and the red one. Defining when the growing behavior will be abnormal to the expected scenario. The correct classification and the accuracy of curvature estimates, measured over variations of significant segmentation variables was used in other structure segmentation techniques in the researched field [42].

Hybrid combination for automatic detection of the seeds within the salt bodies: One of the important steps that are required in our algorithm, is to detect the seeds within the salt bodies, since all the followed steps will be build upon it. In addition to the fact that it will make our solution more global, reliable and user friendly to the interpreter. We suggest in our work, two different approaches. The first is to iterate over all the pixels in the image and recognize if it is a potential seed. The second, where we approximate circular

Figure 6.1 Monitoring the curvature in our growing method



shapes from the salt structures and define the seeds in the centre. Other new approaches can be discussed and tested.

For future development and improvement of our algorithm, we suggest that running two different methods to get the list of the suggested seeds as output, can be applied to a certainty model and ensure getting accurate results with better quality. In other words, when running the two methods separated on the same input, we will get two lists of the suggested seeds. The next step is to compare between the two lists and pick the ones that are in both lists or close to the error rate that the user predefine. If no matching is found between the lists, a warning message will be given to the user about the quality of the input volume and a manual interaction from the user is needed to define the seeds.

REFERENCES

[1] R. Adams and L. Bischof. Seeded region growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(6):641 –647, jun 1994.

[2] Trond H. Boe Ahmed A. Aqrawi. Imaging structural geology with dip and directional dip. *Geo 2012-Bahrin*, mar 2012.

[3] Trond Hellem Boe Ahmed Adnan Aqrawi and Sergio Barros. Detecting salt domes using a dip guided 3d sobel seismic attribute. *SEG annual meeting- San Antonio*, Sep 2011.

[4] France Albert Tarantola, Institut de Physique du Globe de Paris, editor. *A Strategy For Nonlinear Elastic Inversion of Seismic Reflection Data*, SEG Annual Meeting, Houston, Texas, November 2 - 6 1986. Society of Exploration Geophysicists.

[5] Ahmed Adnan Aqrawi and Trond Hellem Boe. Improved fault segmentation using a dip guided and modified 3d sobel filter. *SEG annual meeting- San Antonio*, Sep 2011.

[6] Peter Bakker. Image structure analysis for seismic interpretation. 2002.

[7] M. Basu. Gaussian-based edge-detection methods-a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32(3):252 – 260, aug 2002.

[8] B. Biondi. *Three dimensional seismic imaging*. Investigations in geophysics. Society of Exploration Geophysicists, 2006.

[9] R.H. Chan, Chung-Wa Ho, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *Image Processing, IEEE Transactions on*, 14(10):1479 –1485, oct. 2005.

[10] Satinder Chopra and Kurt J. Marfurt. Seismic attributes a historical perspective. *Geophysics*, 70(5):3SO–28SO, 2005.

[11] J.J. Clark. Authenticating edges produced by zero-crossing algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(1):43 –57, jan 1989.

[12] Israel Cohen and Ronald R. Coifman. Local discontinuity measures for 3-d seismic data. *Geophysics*, 67(6):1933–1945, 2002.

[13] Davis W. Ratcliff Diamond Geoscience Research Corp. David R. Muerdter, Richard O. Lindsay, editor. *Imaging Under Edges of Salt Sheets: a Ray-tracing Study*, 1996 SEG Annual Meeting, Denver, Colorado, November 10 - 15 1996. Society of Exploration Geophysicists.

[14] M G Orozco del Castillo, C Ortiz-Alemn, R Martin, R vila Carrera, and A Rodrguez-Castellanos. Seismic data interpretation using the hough transform and principal component analysis. *Journal of Geophysics and Engineering*, 8(1):61, 2011.

[15] G. Deng and L.W. Cahill. An adaptive gaussian filter for noise reduction and edge detection. In *Nuclear Science Symposium and Medical Imaging Conference, 1993., 1993 IEEE Conference Record.*, pages 1615 –1619 vol.3, oct-6 nov 1993.

[16] Glenn R. Easley and Flavia Colonna. Generalized discrete radon transforms and applications to image processing. volume 151 of *Advances in Imaging and Electron Physics*, pages 169 – 239. Elsevier, 2008.

[17] Jianping Fan and Ahmed K. Elmagarmid. An automatic algorithm for semantic object generation and temporal tracking. *Signal Processing: Image Communication*, 17(2):145 – 164, 2002.

[18] Jianping Fan, Guihua Zeng, Mathurin Body, and Mohand-Said Hacid. Seeded region growing: an extensive and comparative study. *Pattern Recognition Letters*, 26(8):1139 – 1156, 2005.

[19] Satellite image Google Maps. Satellite image of salt tectonic features, iran, 2001.

[20] 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*, 7(12):1684 –1699, dec 1998.

[21] D.J. Higham and N.J. Higham. *Matlab Guide*. Society for Industrial and Applied Mathematics, 2005.

[22] Trygve Randen Lars Sonneland Schlumberger Stavanger Research Hilde G. Borgos, Thorleif Skov. Automated geometry extraction from 3d seismic data. *SEG Annual Meeting - Dallas, Texas*, Oct 2003.

[23] K.K.S. Ho and J.W.C. Lau. Learning from slope failures to enhance landslide risk management. *Quarterly Journal of Engineering Geology and Hydrogeology*, 43(1):33–68, 2010.

[24] Michael R. Hudec and Martin P.A. Jackson. Terra infirma: Understanding salt tectonics. *Earth-Science Reviews*, 82(12):1 – 28, 2007.

[25] Yong Lee and S. Kassam. Generalized median filtering and related nonlinear filtering techniques. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(3):672 – 683, jun 1985.

[26] A. Frank Linville and Robert A. Meek. A procedure for optimally removing localized coherent noise. *Geophysics*, 60(1):191–203, 1995.

[27] Y. Lu and R.C. Jain. Behavior of edges in scale space. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(4):337 –356, apr 1989.

[28] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980.

[29] J.S. Monroe, R. Wicander, and R.W. Hazlett. *Physical geology: exploring the Earth*. Available Titles CengageNOW Series. Thomson Brooks/Cole, 2006.

[30] M.S. Nixon and A.S. Aguado. *Feature Extraction & Image Processing*. Academic Press. Academic, 2008.

[31] Delft University of Technology. Seismic imaging, 2012.

[32] J. Princen, J. Illingworth, and J. Kittler. Hypothesis testing: a framework for analyzing and optimizing hough transform performance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(4):329 –341, apr 1994.

[33] Yong Rui, Thomas S. Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39 – 62, 1999.

[34] Schlumburger Information Solution. *Petrel Introduction G&G*. Schlumburger, 2010.

[35] Schlumberger Information Solutions. Petrel geophysical software, 2011.

[36] Schlumberger Information Solutions. Petrel seismic interpretation, 2011.

[37] T.P.H. Steeghs. Local power spectra and seismic interpretation.

[38] A. Aqrawi T. Boe. Fault detection by local enhancement of statistically significant linear features. *SEG annual meeting-Las Vegas*, NOV 2012.

[39] M. T. Taner, F. Koehler, and R. E. Sheriff. Complex seismic trace analysis. *Geophysics*, 44(6):1041–1063, 1979.

[40] Dag Hermansen Fugro Seismic Imaging Thomas Elboth, Fugro Geoteam, editor. *Attenuation of Noise In Marine Seismic Data*, SEG Annual Meeting, Houston, Texas, October 25 - 30 2009. Society of Exploration Geophysicists.

[41] North Seattle Community College Tom Braziunas. Earthquakes, geologic dating cross-sections, 2001.

[42] E. Trucco and R.B. Fisher. Experiments in curvature-based segmentation of range data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(2):177 –182, feb 1995.

[43] Stephen A. Nelson-Tulane University. Deformation of rock, 2003.

[44] N. Wiener. *The Fourier Integral and Certain of Its Applications*. Cambridge Mathematical Library. Cambridge University Press, 1988.

[45] W.S. Kowalik Chevron Petroleum Technology Company La Habra CA 90633 USA Y. Luo, W. G. Higgs. Edge detection and stratigraphic analysis using 3d seismic data. Denver, Colorado, November 10 - 15 1996. Society of Exploration Geophysicists.

[46] Ö. Yilmaz and S.M. Doherty. *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data*. Number v. 1 in Investigations in geophysics. Society of Exploration Geophysicists, 2001.

[47] Ian T. Young and Lucas J. van Vliet. Recursive implementation of the gaussian filter. *Signal Processing*, 44(2):139 – 151, 1995.

GLOSSARY

Convolution	*is the most important and fundamental concept in signal processing and analysis. By using convolution, we can construct the output of system for any arbitrary input signal, if we know the impulse response of system.*
Cross Line	*A seismic line within a 3D survey perpendicular to the direction in which the data were acquired. (y,z) in 3D xyz.*
Depth Slice	*Equivalent to time slice.*
Dip	*The magnitude of the inclination of a plane from horizontal. True, or maximum, dip is measured perpendicular to strike. Apparent dip is measured in a direction other than perpendicular to strike. It refer to the orientation or attitude of a geologic feature.*
Geophysics	*is the physics of the Earth and its environment in space. The study of the Earth using quantitative physical methods, includes: Earth's shape; its gravitational and magnetic fields; its internal structure and composition; its dynamics and their surface expression in plate tectonics, the generation of magmas, volcanism and rock formation.*
Hybrid filters	*Switching between different filters on the same image during the same run, based on a condition applied to the target pixel.*
In-Line	*A seismic line within a 3D survey parallel to the direction in which the data were acquired (x,z) in 3D xyz. In marine seismic data, the in-line direction is that in which the recording vessel tows the streamers.*
Overburden	*Rock overlying an area or point of interest in the subsurface.*
Petrel	*Software to create powerful seismic to simulation workflows in a familiar Windows environment.*
Post-Stack	*This term is used in Oil and Gas exploration/development to describe seismic inversion. Seismic inversion is the process of transforming seismic reflection data into a quantitative rock-property description of a reservoir to differentiate geologic features.*
Salt body	*A soft, soluble evaporate mineral commonly known as salt or rock salt. Because salt is less dense than many sedimentary rocks, it is relatively buoyant and can form salt domes, pillars or curtains by flowing and breaking through or piercing overlying sediments, as seen in the Gulf of Mexico and the Zagros fold belt.*
Salt Flanks	*Not part of the salt body, it is stratigraphy terminating at the salt body boundaries. It is of special interest for potential oil and gas deposit.*
Salt tectonics	*is concerned with the structural geometries and deformation processes associated with the presence of significant thicknesses of rock salt within a sequence of rocks. This is due both to the low density of salt, which does not increase with burial, and its low strength.*

Sediment	*a naturally occurring material that is broken down by processes of weathering and erosion, and is subsequently transported by the action of wind, water, or ice, and/or by the force of gravity acting on the particle itself. It contains different kind of rocks.*
Seeded growing	*an approaches to segment an image using one point called seed mark, it will grow within a region. Usually, it is described as a seeded region growing.*
Seismic Attribute	*is a quantity extracted or derived from seismic data that can be analyzed in order to enhance information that might be more subtle in a traditional seismic image, leading to a better geological or geophysical interpretation of the data.*
Seismic data	*A set of numerous closely-spaced seismic lines that provide a high spatially sampled measure of subsurface reflectivity. This will provide detailed information about fault distribution and subsurface structures.*
Seismic imaging	*directs an intense sound source into the ground to evaluate subsurface conditions and to possibly detect high concentrations of contamination. Receivers called geophones, analogous to microphones, pick up "echoes" that come back up through the ground and record the intensity and time of the "echo" on computers. Data processing turns these signals into images of the geologic structure.*
Seismic interpretation	*In geophysics, analysis of data to generate reasonable models and predictions about the properties and structures of the subsurface. Interpretation of seismic data is the primary concern of geophysicists.*
Smoothing filters	*are used to enhance noisy images (at the expense of blurring). This filter generates the average over a 3 x 3 area of the image. The technique is also called moving window averaging.*
Stratigraphic	*The study of the history, composition, relative ages and distribution of strata, and the interpretation of strata to elucidate Earth history. The comparison, or correlation, of separated strata can include study of their lithology, fossil content, and relative or absolute age, or lithostratigraphy, biostratigraphy, and chronostratigraphy.*
Texture	*in geology, refers to the physical appearance or character of a rock, such as grain size, shape, arrangement, and pattern at both the megascopic or microscopic surface feature level.*
Time Slice	*A horizontal display or map view of 3D seismic data having a certain arrival time, as opposed to a horizon slice that shows a particular reflection. A time slice is a quick, convenient way to evaluate changes in amplitude of seismic data.*

Appendix A

SHORT PAPER FOR SEG ANNUAL MEETING TEXAS 2013

This is an extended abstract (short paper), written to be sent to SEG annual meeting in Texas 2013 confernce in USA. The actual paper will be written later as the deadline is in September.

Automated Seeded Growing Method for Salt Body Delineation

Nour H. Alabbasi, University of Stavanger (UiS), and*
Ahmed A. Aqrawi, Schlumberger Norway Technology Center.
Stavanger, Norway

Summary

For a more accurate geological model, it is crucial to have proper segmentation and delineation of structures. Salt structures are known to be difficult to segment given their chaotic nature; however, there are methods that isolate the salt borders well. We propose an automated seeded growing method that will segment the salt bodies from the other structures in seismic. This method builds upon the conventional seeded growing approach, and expands it with hybrid smoothing and discontinuous boundary detection. Our results are illustrated on datasets from the Gulf of Mexico, where we see a clear segmentation of the salt bodies and a termination at the flanks.

Introduction

Delineation of salt structures in post stack seismic data is an inherently difficult problem, and is of great value when detecting potential reservoirs. There exist several approaches where an edge detecting seismic attribute is used to detect such structures; such is the case with well known attributes in the coherence family (Chopra and Marfurt, 2007). These attributes usually are a good indication of the flanks of the salt structure, but struggle in segmenting the salt body as a whole. We suggest a method of automated seeded growing to aid in salt body detection. The main aspects are that we automatically detect salt features and place seeds within them, account for noise reduction/removal during growing, salt boundary detection, and incomplete boundary estimation and termination. As input to our method we consider filtered seismic data with typical edge detection seismic attributes.

Seed growing is a known method for detection of structures in seismic (Adams and Bischof, 1994), and has mostly been used for surface detection. We have expanded this with the use of other filtering methods as the growing takes place to target the delineation of salt bodies. We have relied on smoothing and noise reduction filters such as the median and the mean (Lee and Kassam, 1985) (Chan et.al, 2005) (Linville and Meek, 1995) as well and edge/boundary detection methods like amplitude contrast (Aqrawi and Boe, 2011). Another aspect that is of interest when detecting these salt bodies is the discontinuous nature of their boundaries and how to terminate to avoid growing into other structures (Cohen and Coifman, 2002). We have also built further upon this by adding some intelligence to the algorithm where is will decide when to use certain method to optimize the growing accuracy.

To validate our method we have been looking at datasets with salt diapirs from the Gulf of Mexico (GOM). Here we aim at using seismic attribute volumes as input , and as such have run amplitude contrast (Aqrawi and Boe, 2011) and dip illumination (Aqrawi and Boe, 2012) to segment the flanks of our salt bodies before running the seeded growing method. (See Figure 1)

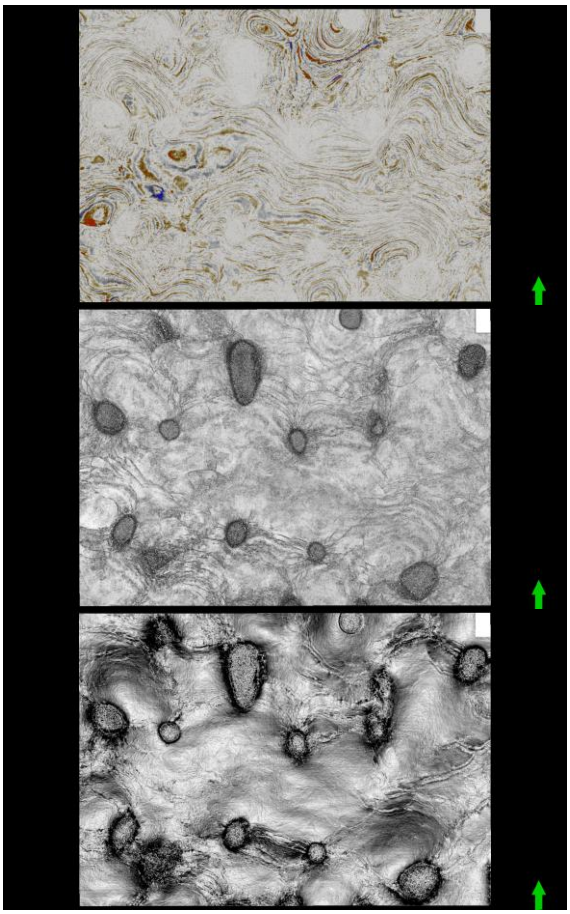


Figure 1: Original input seismic data (top) amplitude contrast filtered volume (middle) dip illumination filtered volume (bottom). (Data courtesy of WesternGeco)

Automated Seeded Growing Method for Salt Body Delineation

Method

Using the edge volumes as input we get a good indication of the flanks and borders of our salt structures. We chose a seeded growing approach for the segmentation because this way we can rely on either the users input as to the positioning within the salt body, or our automated approach. Once a point is identified within the salt body, we can start growing from this seed point and identifying the salt structure. Our automated approach identifies circular shapes, as is commonly the case for salt structures on a time slice, and seed at the center of these circles to start growing. We have already shown that this is possible by using our edge enhancement techniques similar to the Radon transform (Boe and Aqrawi, 2012), but rather now we are identifying circular shapes rather than edges (or line segments). This means that the seeds do not necessarily need to be inputted by the user.

As we are growing within a salt body, there will be the challenge of identifying and distinguishing boundaries (salt borders) from noise. Given the noisy texture of salt data in seismic, we decided to precondition our growing seeds and their neighboring samples with the use of smoothing filters such as and not limited to median, mean, Gaussian, adaptive median, etc. We have also taken into account when to use the various filters for best results and suggest to switch between them, in real time, given an analysis of how chaotic the considered neighborhood is. This means that we are using a hybrid combination of smoothing methods, and our alternating criteria, in this case, is the amount of chaos detected.

As the noise is reduced or filtered away, we then run a normalized Sobel algorithm for edge detection. This is to detect if we are near any salt boundaries and to terminate our growing in this direction. One is not limited to using the Sobel algorithm here, but can also use others in the coherence family of attributes for edge detection. However, we chose this approach due to recent discoveries made by (Aqrawi and Boe, 2011) when comparing the Sobel to coherence attributes for salt detection.

Finally, another unfortunate characteristic in salt data is that it may have discontinuous edges and hence a growing algorithm will grow beyond the salt structure. Our attempt to address this involves estimating the boundary along discontinuous parts of the structure and terminate accordingly. There are several approaches that could be used for the termination estimation, such as using the double derivative and edge enhancement techniques. We have focused on evaluating the growing directions of our seed points and terminate based on unbalanced behavior.

Results & Analysis

Since there are distinct steps in our approach, the results will demonstrate the effects these have on our method. As a first step we can see the results of the automated detection of the seeded within the salt bodies (see figures 2,3). The figure shows that the detection and seeding are all within the salt bodies such that we are only segmenting them. This is important to avoid growing outside of the wanted structures.

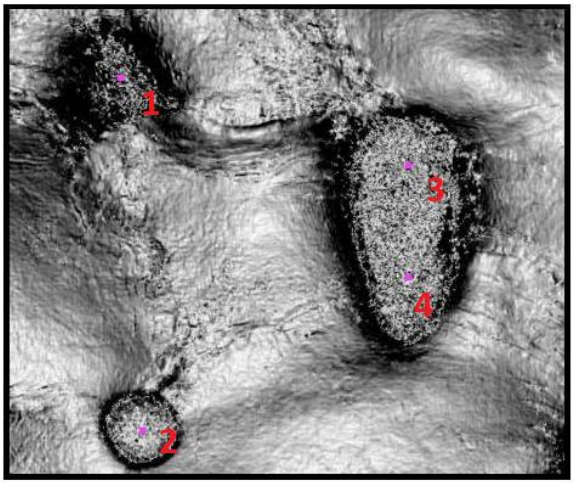


Figure 2: Automated detection of seed points are marked in red. The background image is the input volume to our segmentation method. (Data courtesy of WesternGeco)

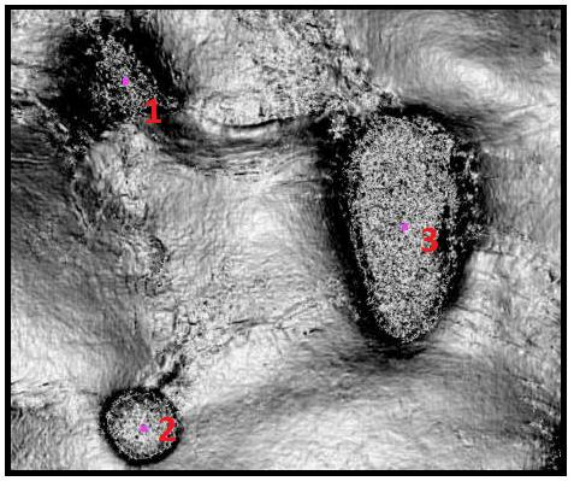


Figure 3: Automated detection of seed points are marked in red. The background image is the input volume to our segmentation method. (Data courtesy of WesternGeco)

In figure 4, an illustration of the detected boundaries of our example datasets are shown. Here we can see a clear detection of the salt bodies, where the flanks are not

included. The results show that even given the noisy nature of the salt images, the method is able to segment the salt bodies entirely and consistently. This is due to our approach of hybrid alternative smoothing. In figure 4, we show the effects of using our hybrid smoother rather than conventional smoothing methods. Here we are comparing between the Gaussian, median and hybrid approaches to smoothing, and it is clear that the hybrid is doing a better job at reducing or removing chaotic features to enhance our boundary detection.

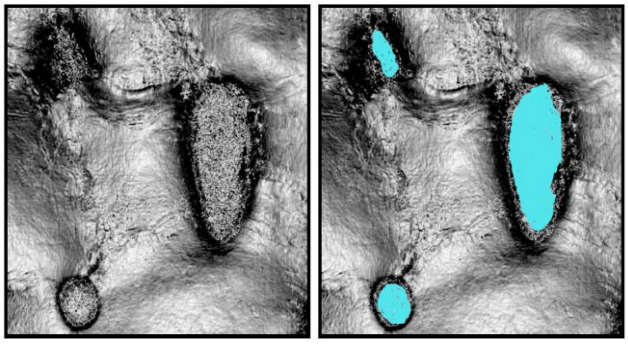
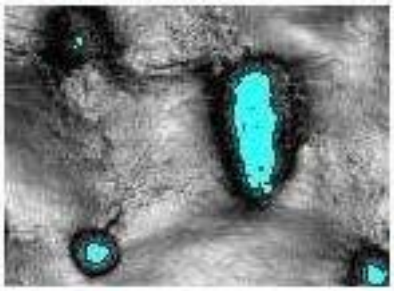


Figure 3: Results of final segmentation on a sub section of the cube (Data courtesy of WesternGeco)

Having reduced noise levels in the growing region, a termination condition needs to be applied such that the growing will stop at the flanks of the salt bodies. Our results (see Figure 4) show that we are terminating only at the boundaries of the salt bodies and not before or after. This gives a more consist segmentation. This is evident even in the case where the structures boundaries are disconnected. Our approach of estimating the boundaries when not present and looking at unbalanced growing, results in a consistent segmentation nonetheless.



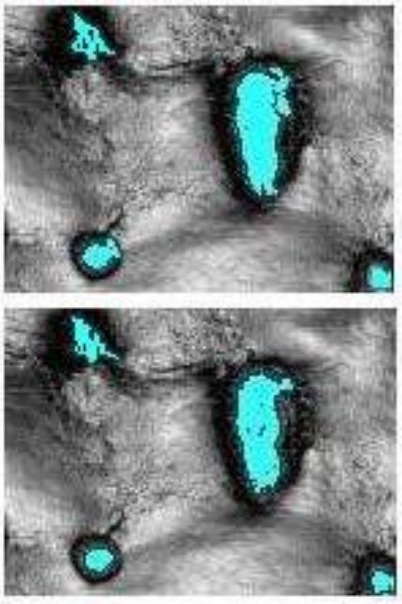


Figure 4: Results of final segmentation given various smoothing methods. Median (top) Mean (middle) Hybrid (bottom) (Data courtesy of WesternGeco)

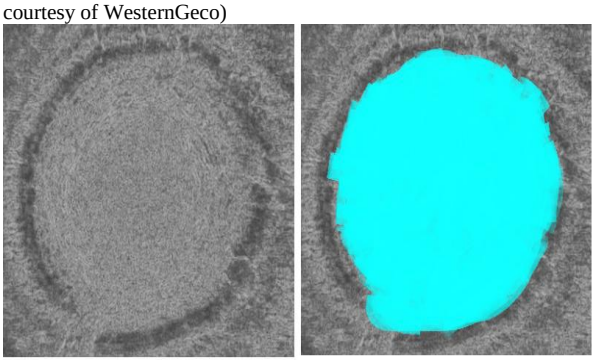


Figure 5: Results of final segmentation and boundaries estimation **(**Data courtesy of WesternGeco)

Conclusions

The approach of a seeded growing algorithm to segment salt bodies proves to be useful, and given a proper input volume is able to isolate the salt body as a whole. The approach presented here, with the modifications, is effective at this task only when combined with methods of noise removal/reduction and boundary termination estimation for existing and discontinuous boundaries. The hybrid smoothing approach has also proven to be very useful combination with the growing method as it alternates smoothing techniques to optimize the segmentation.

Acknowledgements

We would like to acknowledge the Petrel geophysics team for the use of Petrel to test out this method. Also we would like to thank Western Geophysical for the seismic data.

References (submitted separately)

S. Chopra and K. Marfurt, *Seismic attributes for prospect identification and reservoir characterization*. 2007.

Aqrawi, A. and Boe, T.H. *Detecting salt domes using a dip guided 3D Sobel seismic attribute*, SEG Expanded Abstracts 30, 999. 2011.

R. Adams and L. Bischof. *Seeded region growing. Pattern Analysis and Machine Intelligence*, IEEE Transactions on Image Processing, 16(6):641-647, Jun 1994

R.H. Chan, Chung-Wa Ho, and M. Nikolova. *Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization*. IEEE Transactions on Image Processing, 14(10):1479-1485, Oct. 2005.

Israel Cohen and Ronald R. Coifman. *Local discontinuity measures for 3-d seismic data*. *Geophysics*, 67(6):1933-1945, 2002.

Lee, Y., and S. Kassam, 1985, *Generalized median filtering and related nonlinear filtering techniques*: IEEE transactions on Acoustics, Speech and Signal Processing, 33, 672-683.

Linville, A. F. and R. A. Meek, 1995, *A procedure for optimally removing localized coherent noise*: Geophysics, 60,191-203.

A. Aqrawi and T. Boe, *Imaging structural geology with dip and directional dip*, AAPG GEO 2012

T. Boe and A. Aqrawi, *Fault detection by local enhancement of statistically significant linear features*, SEG annual meeting, Nov 2012

Appendix B

PATENT MEMO FOR OUR SEEDED GROWING METHOD

This is a provisional patent filed to protect our work in the next year. It was filed in 29.05.2012 with a patent number: IS122633USPSP_20120529.

Electronic Acknowledgement Receipt

EFS ID:	12886781
Application Number:	61652793
International Application Number:	
Confirmation Number:	1943
Title of Invention:	SEEDED GROWING ALGORITHM FOR SALT BODY SEGMENTATION IN POST STACK SEISMIC DATA
First Named Inventor/Applicant Name:	Nour Hafiz Alabbasi
Customer Number:	48879
Filer:	Cuong L. Nguyen/Bruce Flanders
Filer Authorized By:	Cuong L. Nguyen
Attorney Docket Number:	IS12.2633-US-PSP
Receipt Date:	29-MAY-2012
Filing Date:	
Time Stamp:	20:27:25
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$250
RAM confirmation Number	8295
Deposit Account	071078
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Provisional Cover Sheet (SB16)	IS122633USPSP_20120529_Provisional-Application-Cover-Sheet-as-Filed.pdf	2071747 62ca455ed00c558bb04e9aee6701deba4f663b43	no	3
Warnings:					
Information:					
2		IS122633USPSP_20120529_PSP-as-Filed.pdf	2863982 3f6105d9f630b740e4f95ce036c1fc451e87ecb0	yes	12

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Specification	1	10
	Claims	11	11
	Abstract	12	12

Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	29901 233c31f75e5010d0a5a150b5a353177242631f99	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	4965630	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Appendix C

POSTER UIS 2012

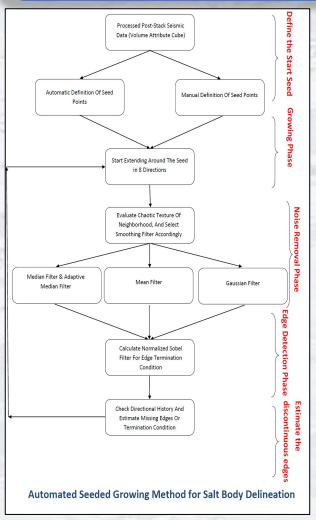
Seeded Growing Algorithms for Salt Body Segmentation in Post-Stack Seismic Data

Nour Alabbasi
University of Stavanger-Department of Electrical and Computer Engineering

SUMMERY

For a more accurate geological model, it is crucial to have proper segmentation and delineation of structures. Salt structures are known to be difficult to segment given their chaotic nature; however, there are methods that isolate the salt borders well. We propose an automated seeded growing method that will segment the salt bodies from the other structures in seismic data. This method builds upon the conventional seeded growing approach, expands it with hybrid smoothing and discontinuous boundary detection. Our results are illustrated on datasets from the Gulf of Mexico, where we see a clear segmentation of the salt bodies and a termination at the flanks.

METHODOLOGY



Automated Seeded Growing Method for Salt Body Delineation

To summarize our work we can say that we have 4 unique steps:

•The use of the salt structures and the combination of image processing techniques such as smoothing, edge detection and boundary estimation within the growing algorithm is unique.

•To automate the seed picking such that user does not need to manually choose the seeds. This is done through an edge enhancement technique similar to edge evidence but focuses on circular shapes rather than line segments.

• The use of several noise removal filters in a hybrid approach and switching between them according to the neighborhood filtered.

•Our approach for non-complete boundaries and how we use directional history for the seeding points to determine termination along a non-existing border.

RESULTS

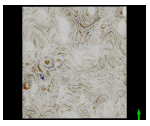


Figure 1

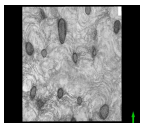


Figure 2

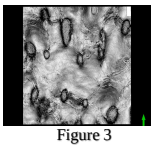


Figure 3

In Figures (1,2,3), we can see that using the edge volumes as input we get a good indication of the flanks and borders of our salt structures.

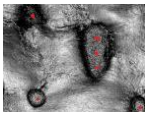


Figure 4

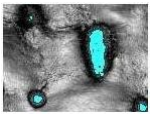


Figure 5

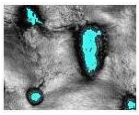


Figure 6

As a first step, we can see the results of the automated detection of the seeded within the salt bodies (Figure 4). The Figures (5,6) show that the detection and seeding are all within the salt bodies. The Figures represent using different smoothing filters such as Median filter (Figure 5) and Mean filter (Figure 6).

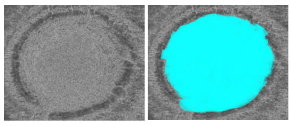


Figure 7

The results in figure 7 show that we are terminating only at the boundaries of the salt bodies and not before or after. This gives a more accurate segmentation. This is evident even in the case where the structures boundaries are disconnected.

CONCLUSION

The approach of a seeded growing algorithm to segment salt bodies proves to be useful, and given a proper input volume is able to isolate the salt body as a whole. The approach presented here with the modifications, is effective at this task only when combined with methods of noise removal/reduction and boundary termination estimation for existing and discontinuous boundaries. The hybrid smoothing approach has also proven to be a very useful combination with the growing method as it alternates smoothing techniques to optimize the segmentation.

Schlumberger