SEISMIC FEATURES DETECTION

A project Sponsored by LMKR

PROJECT SUPERVISOR

Brig. M. Minhas Khan

SYNDICATE MEMBERS:

NC Fareeha Ikram

NC Zaiba Shah

NC Urva Latif

NC Muhammad Sarmaad Masood

Dated: 19th May, 2016

BETE-49

ABSTRACT

Manual data inspection and seismogram interpretation requires processing for event detection, signal classification and data visualization. The use of machine learning techniques automates decision processes and reveals the statistical properties of data. This approach is becoming more and more important and valuable for large and complex seismic records. Unsupervised learning allows the recognition of features. Self-Organizing Maps (SOMs) are used for a data-driven feature selection, visualization and clustering of attributes. The aim of the project is to design an automatic method which clusters the attribute volumes of seismic images to segment different types of features using self-organizing maps.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

- SOMs   Self-Organizing Maps
- CPU   Central Processing Unit
- GPU   Graphical Processing Unit
- Z-Score   Zero-Mean Normalization
- BMU   Best Matching Unit

INTRODUCTION:

1.1. Overview:

Manual data assessment and seismogram interpretation requires processing for event recognition, data visualization and signal classification. The use of machine learning techniques automates decision processes and reveals the statistical properties of data due to which it is becoming common. The recognition of features in the complex seismic volume can be achieved by unsupervised learning. The aim of the project is to design an automatic method which clusters the attribute volumes of seismic images using self-organizing maps for feature identification.

1.2. Problem Statement:

Visualization and analysis of multiple attributes of seismic data set for oil exploration requires manual inspection which is a tedious process and it may lead to error. Oil digging companies have invested a huge amount of money and false decisions might lead to financial loss.

1.3. Objectives:

Academic Objectives:

Techniques of artificial intelligence and machine learning are used in the project.

Application Objectives:

To form an algorithm which automatically clusters various attributes of seismic images to identify different types of features.

1.4. Approach:

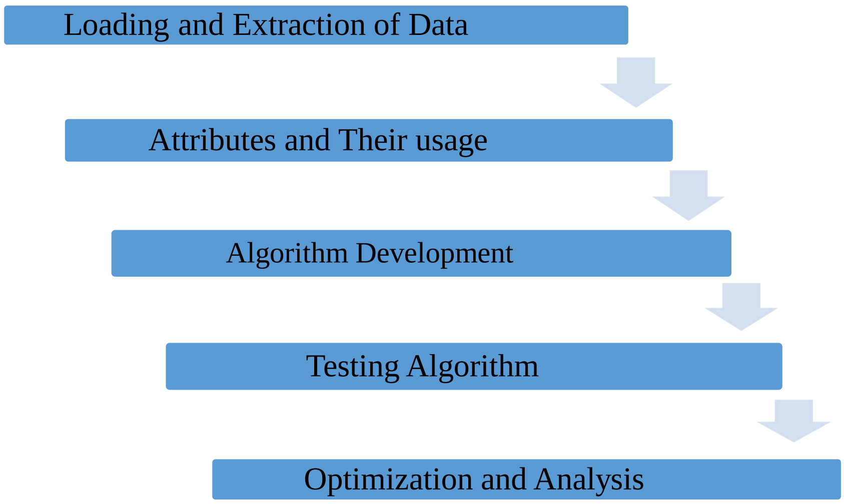


Figure 1. Basic Approach

2. BACKGROUND STUDY:

Large 3D volumes of the layers of the subsurface are called seismic images. Bakker [1] describes that the introduction of 3D seismic images caused a revolutionary change as it helped in choosing the view angle freely. A horizontal view or a time slice is very useful for the interpretation of depositional structures. These interpretation play a vital role in our project to get the desired results[1].

The Seismic images' attributes show some properties of the seismic volume. They are usually computed from the volume itself. Taner. et.al [2] explain the application of complex trace analysis to seismic data and its suitability in geologic interpretation.

Seismic attributes are used to pinpoint features of the subsurface rocks such as faults, slope, frequency, etc. Due to seismic facies analysis seismic feature detection is important[2].

Smith [3] describes the use of SOM for Seismic facies analysis in an effective manner. Furthermore, the primary survey of the reflection amplitude is interpreted along with 4-25 derived attributes. These attributes are fed to artificial neural networks so that they can be organized in a human readable form[3].

3. DESIGN AND DEVELOPMENT:

3.1. Design requirements:

i. Opendtect version 5
ii. MATLAB version 7 or above
iii. Operating system--Windows 7 or above
iv. PC with processor i5 or above and RAM minimum of 8GB
v. NVIDIA GPU minimum Compute capability:'2.1' and total memory:'2.14e+9')

3.2. Technical Specifications:

The project is designed to perform on multiple platform like MATLAB (CPU/GPU) and R to provide geologist a mechanism by which he can easily extract features of his interest in an effective and efficient manner.

Code to these cannot be revealed since they are protected under a Non-Disclosure Agreement (NDA) signed with LMKR, but executable files will be demoed**.**

3.3. Design:

Figure 2 shows the basic design which involves the pre-processing of the input data, then feeding it to SOM to get a clustered image depicting different features.

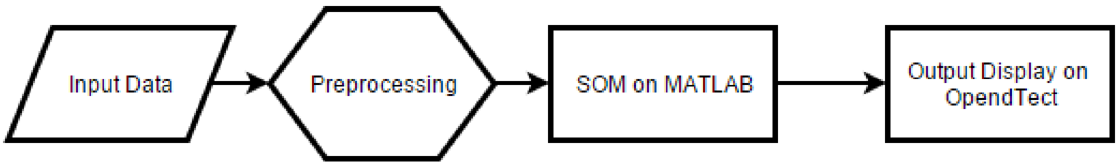


Figure 2.Basic Design of the Project

3.3.1 Preprocessing:

3.3.1.1 Data loading and extraction:

Opedtect is used to load data and generate segy file. Attributes can also be calculated as segy file. These attributes will be given as input to the SOM algorithm.

Seismic Data:

Seismic data used in the project is an open source data known as 'F3 block'. This survey was done at Netherland, offshore North Sea[4].

Opendtect:

Opendtect is software for interpreting seismic data. It is a software system used for visualizing and processing multi-volume seismic data. Large volume can be divided into smallerchunks to interpret seismic data in more detail. It is helpful to understand the seismic images taken for the detection of hydrocarbons underground.

Figure 3 shows the 3D visualization of Seismic data on Opendtect.

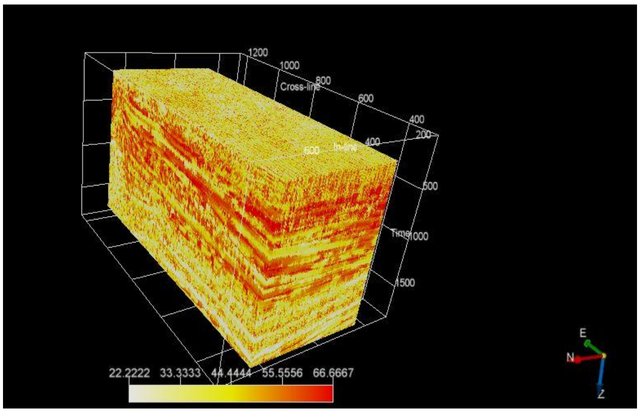


Figure 3. Visualizing Seismic Data in Opendtect

3.3.1.2: Attributes Computation and their usage:

Tensor based attributes are used for attribute calculation. First three Eigen values {λ_1 λ_2 λ_3} and Eigen vectors are calculated for each voxel.

Energy:

Energy is the post stack attribute that gives us the summation of squared amplitudes divided by the total number of samples used within that window[5]. This attribute is used for indication of bright spots which are used as a direct hydrocarbon indicator. Energy is calculated by formula

$$ENERGY = \lambda_1 + \lambda_2 + \lambda_3 \qquad \text{(i)}$$

Figure 4 shows image of energy calculated on a slice '528ms'that has a bright spot represented by black color in the image.

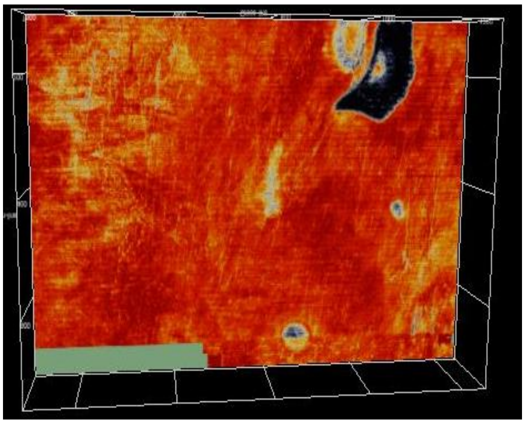


Figure 4 Image of Energy at 528ms time slice

Dip

Dip computes on the horizon, the plane which fits best to its adjacent trace. The output give us the maximum value between the two traces or two planes. Dip is calculated by finding the instantaneous phase for all of the 3 adjacent traces and the dip at the center trace is computed. Dip [6] is useful in that it makes faults more discernible, fault detection. In 3-D data sets by calculating dip across inline and cross line we get maximum change in that trace. Let $\frac{\mathrm{dt}}{\mathrm{dx}}$ and $\frac{\mathrm{dt}}{\mathrm{dy}}$ be the corrected dips in in-line and cross-line directions respectively, then following equation will give the maximum dip:

$$\Delta T = \sqrt{\left(\frac{dt}{dx}\right)^2 + \left(\frac{dt}{dy}\right)^2} \qquad \text{(ii)}$$

Image of Dip calculated on a '528ms' slice is shown in figure 5 which shows maximum value of change in magnitude.

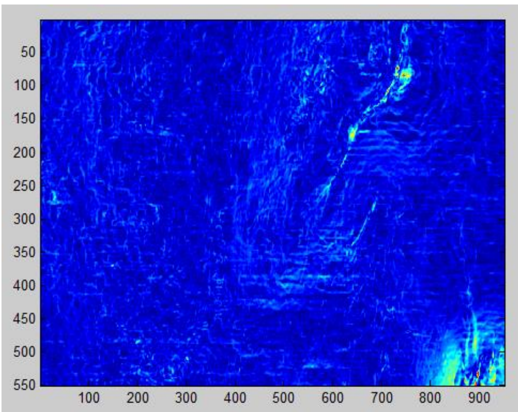


Figure 5. Image of Dip at 528ms time slice

Azimuth

Azimuth is a perpendicular measure to the strike, and is the compass direction (0-360º) of the normal to the plane. Following equation will give azimuth:

$$\Phi = tan^{-1}(\frac{dt}{dx}, \frac{dt}{dy}) \qquad \text{(iii)}$$

The computation is valid for dips up to 180º phase differences. When dip is calculated it gives the maximum value of changes between the traces while this azimuth attribute identifies direction of that maximum change. This attribute is useful as it help us in judging the exact location of the faults.

Image of Azimuth calculated on a '528ms' slice is shown in figure 6 which gives maximum orientation of dip.

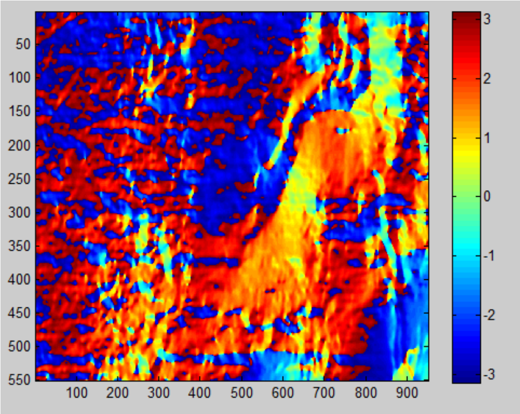


Figure 6.Image of Azimuth at 528ms time slice

Coherency

Coherency [6] gives us the measure of similarity between multiple traces. Coherency convert the data into a volume of discontinuity which help us in revealing fractures, faults and lithological variations. We calculate Coherency by:

$$\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \qquad \text{(iv)}$$

Image of Coherency calculated on a '528ms' time slice is shown in figure 7.

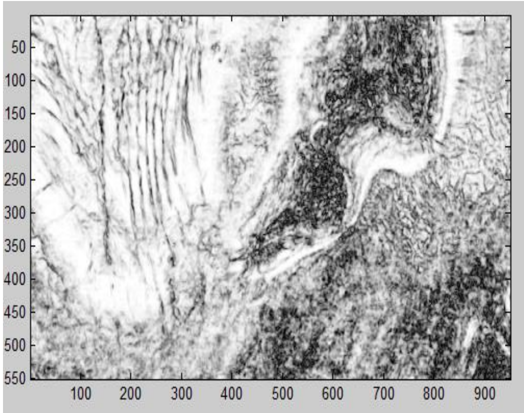


Figure 7. Image of Coherency at 1648ms time slice

Chaos

Chaos attribute gives us the “lack of organization” within the specified window. The low consistency estimated measurement give us chaotic signal pattern and these patterns are used to enhance reflectors, discontinuities, salt bodies, and faults with chaotic texture. We calculated Chaos by

$$\left(\frac{2\times\lambda_2}{\lambda_1+\lambda_2}\right)-1 \qquad \text{(v)}$$

Image of Chaos calculated on a ‘528ms’ slice in shown in figure 8.

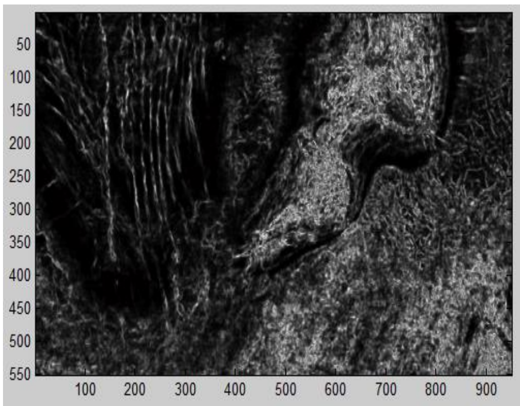


Figure 8. Image of Chaos at 1648ms time slice

Dominant frequency:

Dominant frequency is the concept of selecting dominant frequency and suppress other frequencies. Basically it detects hydrocarbon because where liquid is present then at that place frequency is low. In order to be able to isolate and enhance the salt bodies, high resolution variance we use this attribute.

Image of Dominant frequency calculated on a ‘528ms’ slice is shown in figure 9.

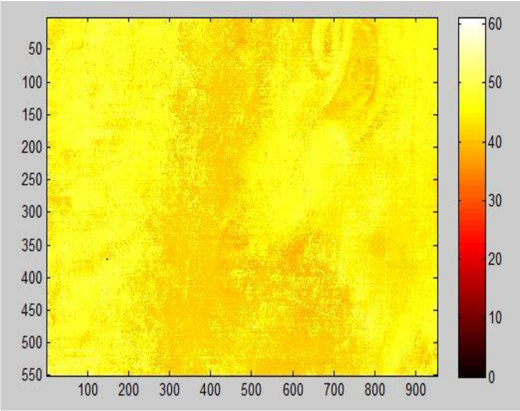


Figure 9. Image of Dominant Frequency at 528ms time slice

Isotropy:

If the vector measurement of layers varies with direction than we get an anisotropy. Similarly if layers are not changing with direction then it is called as isotropy. Two types of anisotropy are found in seismic data:

1. Horizontal Transverse Isotropy
2. Vertical Transverse Isotropy

Image of Isotropy calculated on a '528ms' slice is shown in figure 10.

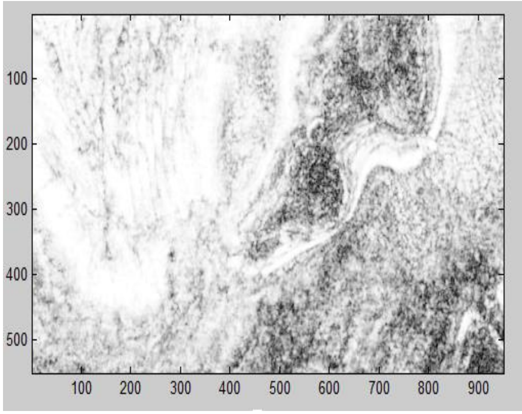


Figure 10. Image of Isotropy at 1648ms time slice

Fault Enhancement:

Fault enhancement attribute is a post stack attribute which is used for analysis of fault geometries that is used to detect turbidities, channel infill because they are not clearly seen on isotropic attribute because of seismic data reflection and it is very necessary for detection of such features because if we do not counter for these attributes than we are increasing the possibility of risk in digging reservoir/well.

Image of fracture fault calculated is shown in figure 11.

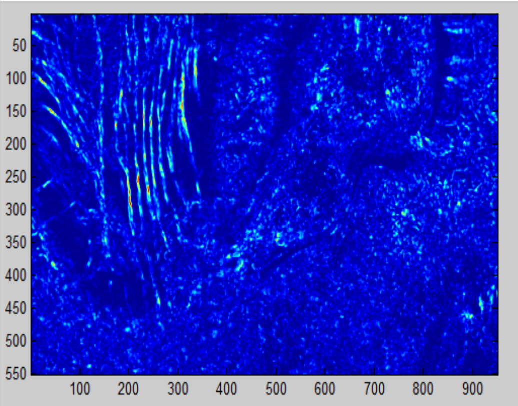


Figure 11. Image of Enhanced Fault at 1648ms time slice

3.3.1.3 Normalization:

It is needed to normalize the attributes before giving them as input to SOM due to following reasons:

1. The input consists of different attributes. All these attributes have their own minima and maxima. Attributes will have different measuring units. All attributes are normalized one by one to bring them in same range [0, 1], then input for SOM is prepared.
2. The neurons are randomly initialed in range [0, 1]. So to compare the input vectors to neurons, the input should be in range similar to that of neurons.
3. The use of heat maps makes it must that input should be in range from [0, 1].

There are many normalization methods that can be followed. If the data has all positive values then simply normalize it by dividing it by max of the data. But as the data used in the project has positive as well as negative values then one of the following methods can be used:

Z-score normalization:

Z-score [7]is also known as zero-mean normalization. The standard deviation and mean is used in normalizing the attribute X. A new value new_v is obtained using the following expression:

$$new\_v = \frac{v - \mu_x}{\sigma_x} \qquad \text{(vi)}$$

where σ_X and μ_X are the standard deviation and mean of attribute X.

As calculating mean and standard deviation for every attribute becomes computationally slow so using this method will not be a good approach.

Normalization by decimal scaling:

In this method of normalization, the decimal point of attribute X values is moved and maximum absolute value of X give the number of decimal points that have to be moved. A normalize value new_v is found by using:

$$new\_v = \frac{v}{10^c} \tag{vii}$$

where c is the smallest integer such that max(|new_v|)<1.

Min-max normalization:

This method performs a linear transformation on the original data values. If minimum and maximum of any feature X is min_x and max_x and we want to map this interval [min_x, max_x] into a new interval [new_min_x, new_max_x] ,then for every value v from the original interval ,we will get new_v by using following formula:

$$new_v = \frac{v - min_x}{max_x - min_x} . (new\_max_x - new\_min_x) + new\_min_x \tag{viii}$$

Min-max normalization preserves the relationships among the original data values.

Min-Max normalization method is used as this method is not heavy on the computation and also different attributes can be assigned different weightages.

Figure-12 shows the normalization of multiple attributes.

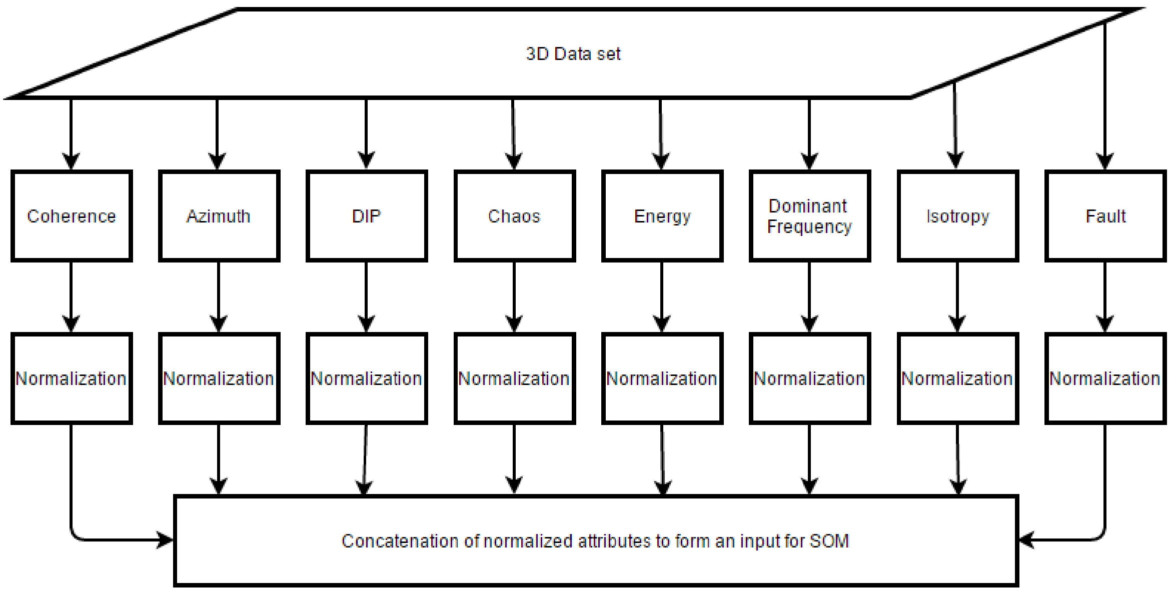


Figure 12.Normalization of Attributes

3.3.2 Training/Learning:

Supervised Learning:

Supervised learning deals with both input vectors and output vectors. The input vectors are trained according to the desired output vectors.

In our case output vectors are unknown so we would be using unsupervised learning methods.

Unsupervised learning:

In unsupervised learning the network try on its own to detect patterns in the input data when the correct match is not known. Input vector is analyzed entirely without human interaction and output vector generated is not used for learning. These features makes it so important for the data mining of large and complex data set[8].

3.3.2.1 Self Organizing Maps:

Self-organizing maps (SOM) is an artificial neural network that uses competitive learning for clustering of the data.

Structure of a SOM:

The structure of a SOM is best understood with the use of an illustration in figure 13

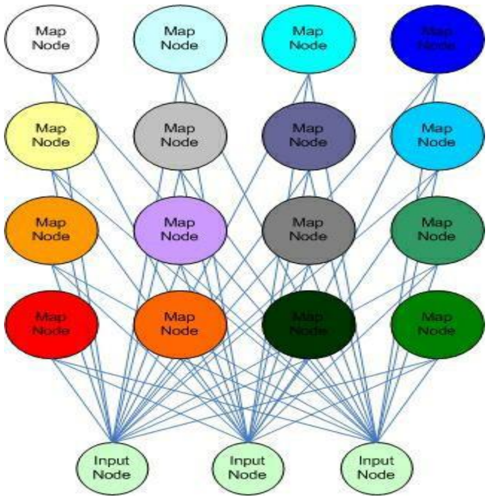


Figure 13. 4x4 SOM network (4 nodes down, 4 nodes across)

SOM consists of:

1. Input layer which consists of randomly distributed input data.

2. Output layer which consists of neurons/nodes.

It has following two features:

1. Each input sample is connected to all the neurons.
2. All the neurons are connected with each other.

All the nodes are organized to form a 2-D grid which makes it easy for visualization of the patterns in the input data. Every node is assigned a unique position (i j) in hexagonal or rectangular grid. This makes it easy to reference a node in the network and to calculate the distances between nodes.

Because of the two features mentioned above whenever an input sample is hit, it act as a stimuli to the map nodes i.e a map node will only update its weights based on what the input vector tells it. So with every iteration the SOM neurons actually try to get as close as possible to the input vector and this is how the map will organize itself.

Importance of SOM:

i. The main advantage of SOM over other clustering techniques is ease of visualization of the results.
ii. In SOM algorithm, number of classes are not defined in the start as compared to K-means.
iii. SOM algorithm results in better output when used on random data as compared to K-means.
iv. The topological neighborhood function is used in SOM for preservation of properties of input data set.
v. Vector quantization is another property of SOM also known as data compression technique. Vector quantization is used to represent high dimensional multi-variant data into lower dimension that is 1 or 2. As humans can easily visualize data in lower dimension than high dimensions so vector quantization of such data helps in their visualization and interpretation.

3.3.2.2 Components of Self Organization:

There are five main components of SOM.

1. **Input:**

The normalized attributes are passed to SOM as shown in figure 14:

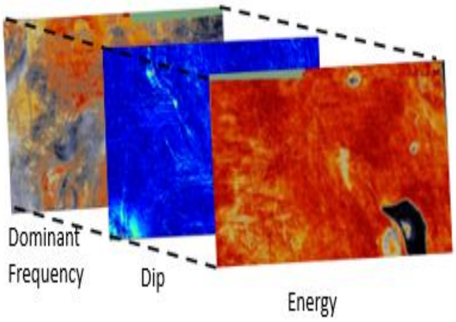


Figure 14.Input to SOM

2. **Initialization**: All the connecting neurons are assigned weights randomly.

3. **Competition:** Neuron compute the discriminant function of input vector which is chosen randomly. Every node in the network is checked for finding a weight which is closest to the input vector selected and the neuron having smallest value is known as the winner or Best Matching Unit (BMU)[8].

$$\text{Distance From Input}^2 = \sum_{i=0}^{n}(I_i - W_i)^2 \qquad \text{(ix)}$$

where I=current input vector

W=node's weight vector

n=number of weight

4. **Co-operation:** The winning neuron regulates the spatial location of a topological neighborhood of excited neurons. So this gives cooperation between neighboring neurons.

$$\sigma(t)= \sigma_0\, e^{(-t/\lambda)} \qquad \text{(x)}$$

where t= current iteration

λ= time Constant

σ_0 = Radius of the Map

σ =Current Radius of the neighborhood

$$\Theta(t)=e^{\wedge}((-\text{distFromBMU})^2/(2\sigma^2(t))) \qquad \text{(xi)}$$

where $\Theta(t)$=Distance from BMU

5. **Adaptation:** By modification in the values of associated connected weights ,the discriminant value is reduce in comparison with input pattern so values of similar input can be increased by selecting the BMU for any preceding application .

$$W(t+1)=W(t)+\Theta(t)L(t)(I(t)-W(t)) \quad \text{(xii)}$$

$$L(t)=L_o\, e^{(-t/\lambda)} \quad \text{(xiii)}$$

where L(t)= Current learning rate

W(t+1)=Updated weight

3.3.3 Voxel Mapping:

After training, each neuron is assigned a unique color on 2D/1D color map. The color of neuron is assigned to each seismic voxel based upon the Euclidean distance of each voxel form the nearest neuron.

3.3.4 Statistical Analysis:

3.3.4.1 Quantization Error:

It is the average distance between each data sample and its best matching unit.

$$\omega^2 = \sum_{i=0}^{n}(I_i - BMU)^2 \quad \text{(xiv)}$$

where ω= Distance between Input and BMU.

I=current input vector

$$\text{Quatization error} = \text{Mean}(\sqrt{\omega^2}) \quad \text{(xv)}$$

Minimum it is, more reliable our results are[9].

It is calculated as one of the goal of SOM is vector quantization due to which this error is generated.

3.3.4.2 Topographic Error:

It is the average distance between the winner neuron and the second best matching unit, for all data samples in terms of (x,y) coordinates in the map[10].

Minimum it is, more reliable our results are.

Its value is between 0-1.

where 0-> topology has been perfectly preserved

1-> topology has not been preserved completely

It is used to check that during training topology of the input data is preserved or not.

3.3.5 Integration with Opendtect:

As our goal was to visualize the output on the Opendtect for better visualization of 3D volume, so output mat files are imported to Opendtect.

Figure-15 shows the full design of the project.

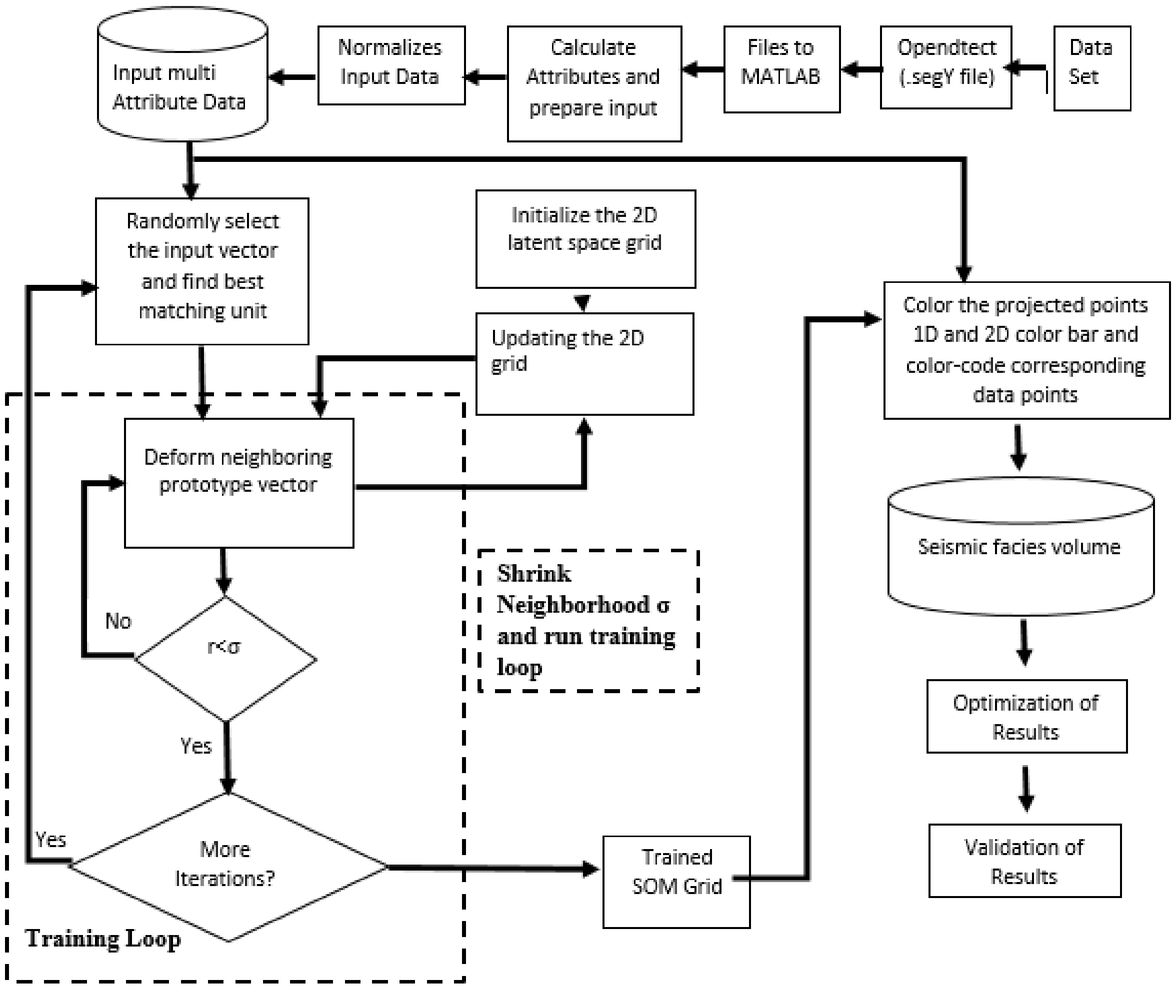


Figure 15. Work Flow of Seismic Feature Detection

3.4 Results and Analysis:

Major analysis of the project is focused upon the detection of the bright spot and fault detection. As mentioned in the section 3.3.1.2, frequency and energy are good measure of hydrocarbons and coherency, isotropy and chaos help in identifying faults. So giving these two attributes as input to SOM would help geologist in identifying the bright spot.

3.4.1 Identifying the known and unknown bright spot:

For the verification of the algorithm, known bright spot in the F3 Block is identified and it is observed whether that bright spot is visible in the clustered image or not.

1. For the analysis the known bright spots are identified by observing different time slices of the F3 block.

   Figure 16 shows that the known bright spot is present at 528ms.

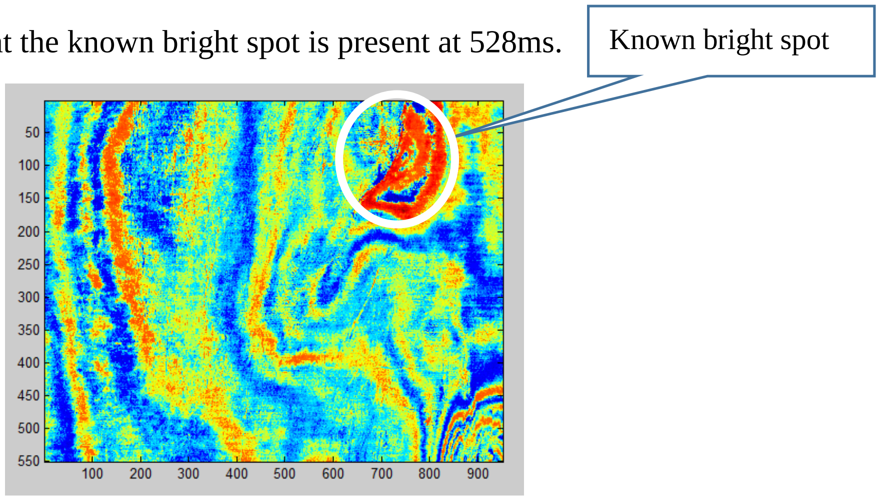


Figure 16. Seismic Image of 528ms time slice

2. Figure 17and Figure 18 depict that another bright spot might be present at the 600ms and 1700ms time slice. So assure it by clustering the 528ms, 600ms and 1700ms slices of energy and frequency.

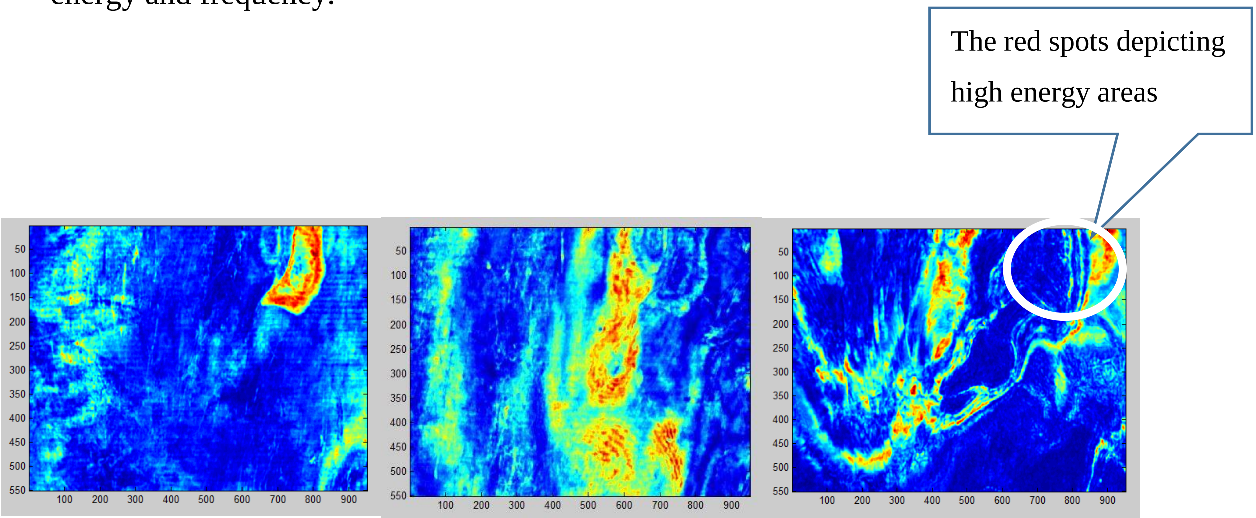


Figure 17. Energy Images for (a) 528ms (b) 600ms (c) 1700ms

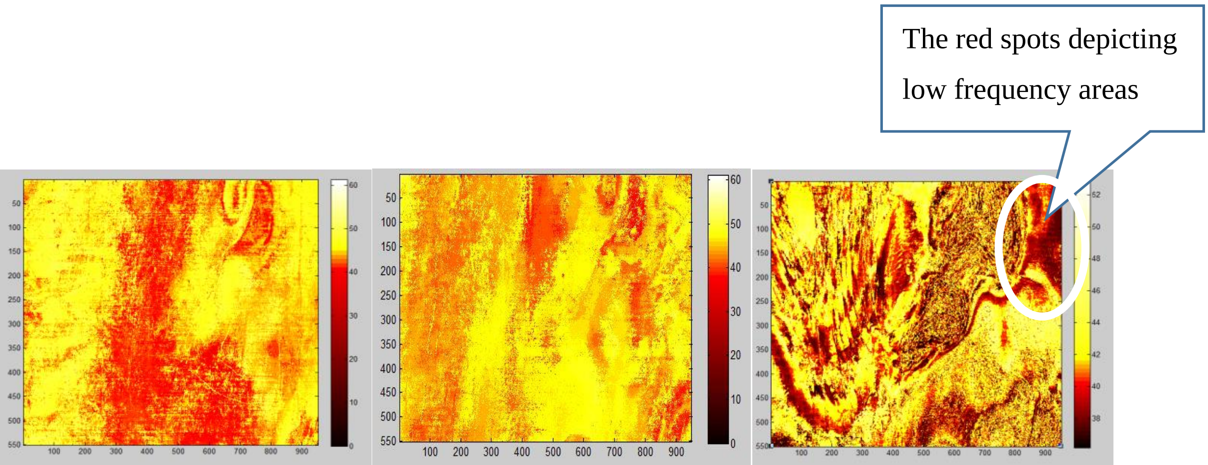


Figure 18. Median Frequency Images for (a) 528ms (b) 600ms (c) 1700ms

3. From the results shown in the Figure 19 it is clearly seen that not only the known bright spot is visible but also the unknown bright spot is identified.

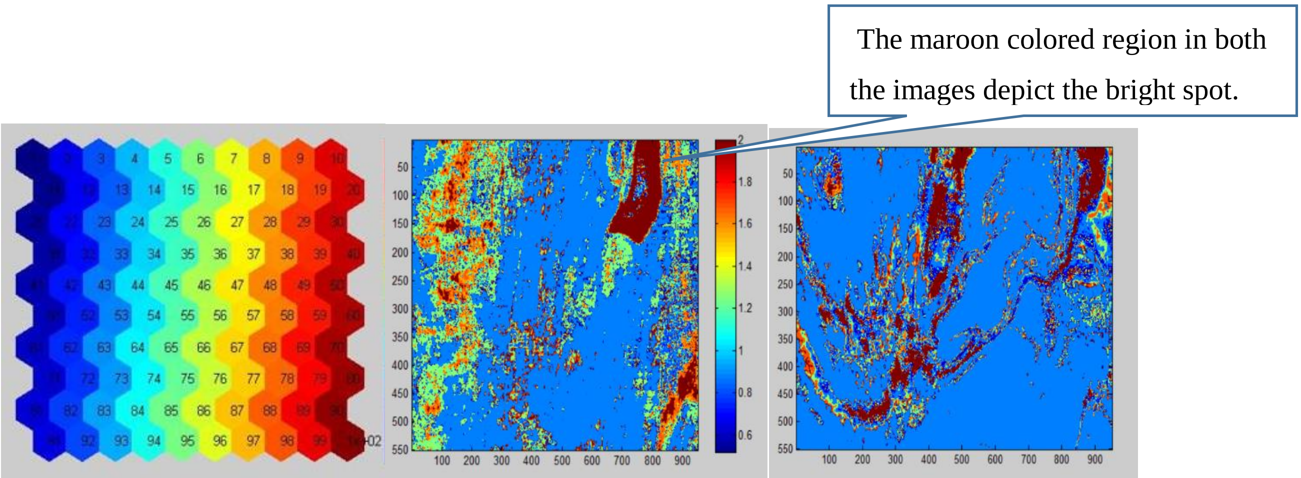


Figure 19. Fused Image generated after clustering of 528ms and 1700ms time slice of energy and frequency

4. From the results shown in the Figure 20 it is clearly seen that not only the known bright spot is visible but also the unknown bright spot is identified.

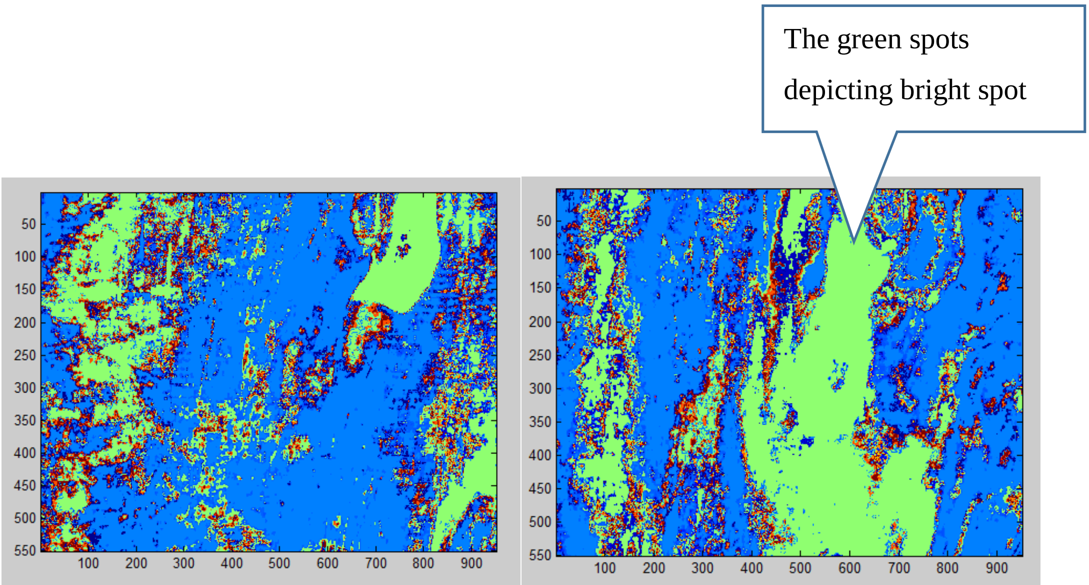


Figure 20.Fused Image generated after clustering of 528ms and 600ms time slice of energy and frequency

3.4.2: SOM Parameters:

In SOM neurons adjust positions recursively so the choice of initial values and the number of time steps is crucial. To study the effect of these parameters on the results is an important part of the analysis.

As mentioned in the section 3.3.2 there are two exponentially decaying functions in the SOM algorithm. These functions should not be decaying too fast or too slow. So the comparison of following parameters is done:

3.4.2.1 Number of Epochs:

The Figure 21 shows that training with 500 iteration won't provide good learning as it would decay too fast and if there is large variability in the input set then the network may not learn very well or at all.

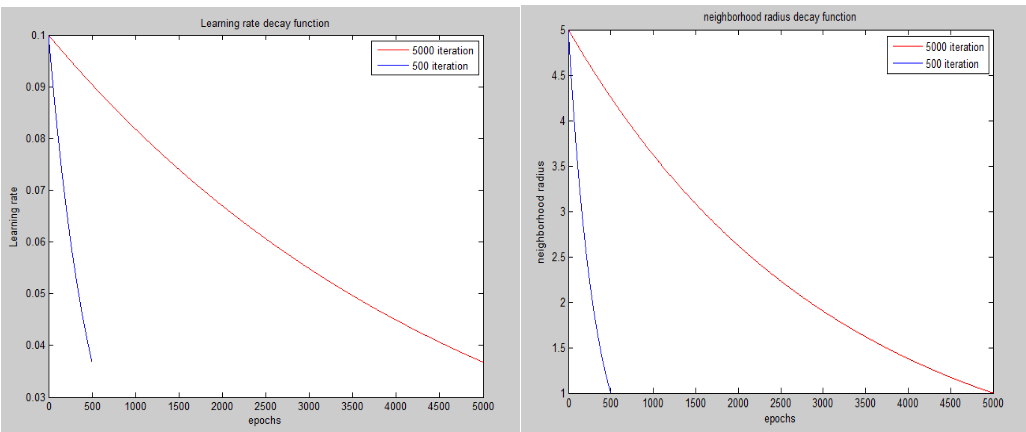


Figure 21. (a). Learning Rate Decay Function (b). Neighborhood Radius Decay Function

Figure 22 depicts that iteration greater than 1000 will produce desirable results.

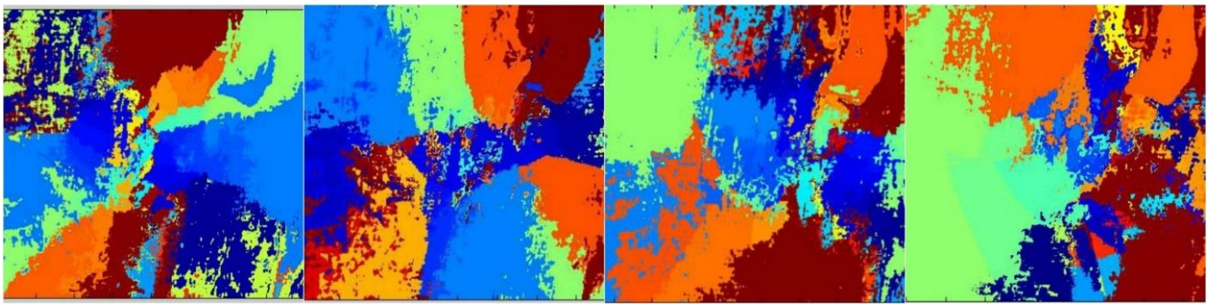


Figure 22. Results for iterations (a) 500 (b) 1000 (c) 5000 (d) 10000

3.4.2.2 Initial Learning Rate:

In Figure 24(a) it can be obesrved that with initial learning rate 0.01 the results are not desireable as the encircled area is fused into the background which is the expected bright spot in the F3 block while in Figure 24(b) for learning rate of 0.1 the encircled bright spot can clearly be identified.

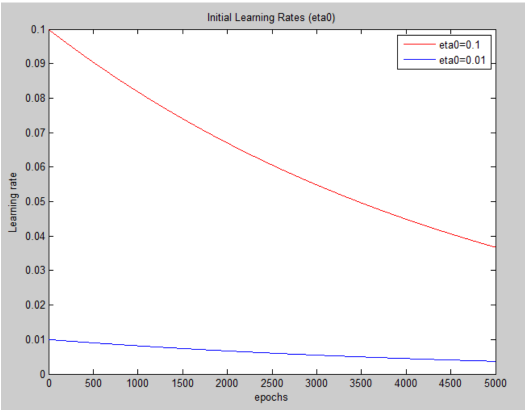


Figure 23. Initial Learning Rates

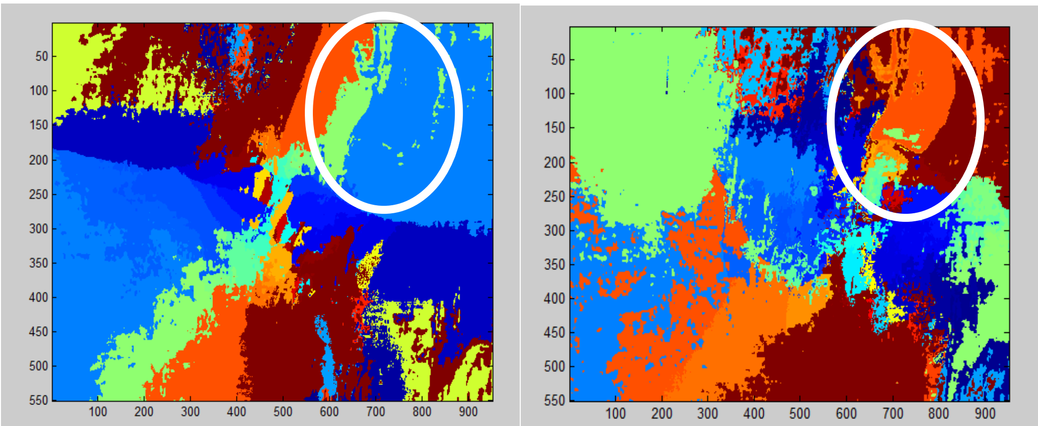


Figure 24. Results for (a) eta0=0.01 (b) eta0=0.1

3.4.2.3 Initial neighborhood Radius:

It can observe from the Figure 26(a) and (b) that for a grid size of 10x10, initial neighborhood radius should be atleast half of the grid size.

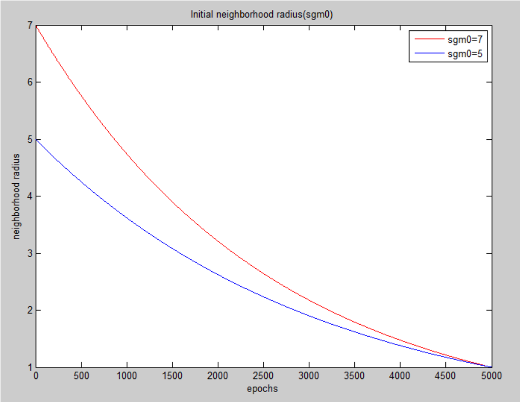


Figure 25. Initial Neighborhood Radius

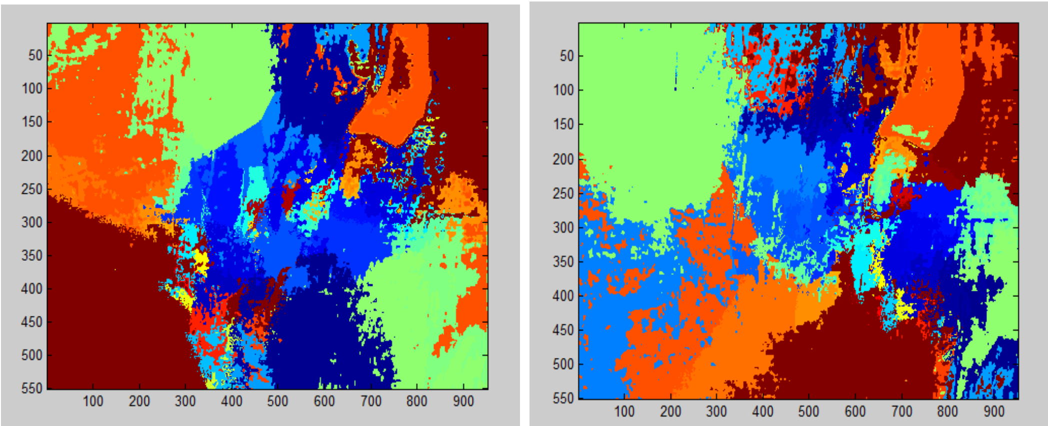


Figure 26.Results for (a) sgm0=7 (b) sgm0=5

3.4.2.4 Grid Size:

Figure 27 (a) Grid size of 5x5 gives well-defined clusters but bright spot is unidentifable.Figure 27 (b) Grid size of 10x10 gives well-defined clusters and bright spot is also identifable while in Figure 27(c) grid size of 15x15 gives unrecognizeble clusters.

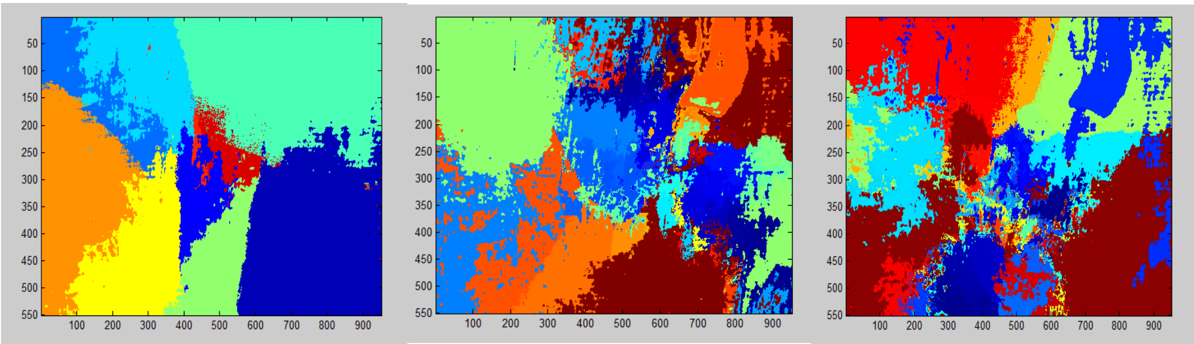


Figure 27. Grid Size (a) 5x5 (b) 10x10 (c) 15x15

3.4.3 Voxels being hit during Training:

Figure 28 (a) gives a fair idea that during training random selection of voxels from the input data covers overall all the in lines and cross lines so that no major chunk of data is missed.

While in Figure 28 (b) it can be seen that voxels selected during training are quite separated from each other, hence major chunk of data in between is missed.

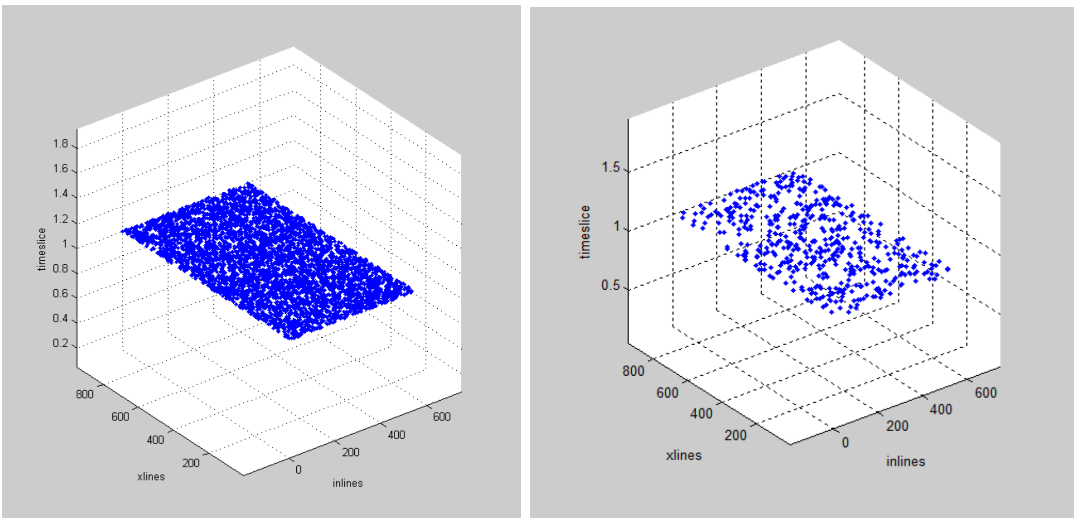


Figure 28. Voxel being hit during Training (a) 5000 iterations (b) 500 iterations

3.4.4 Trained weights:

For every attribute there is a weight/component plane. These helps in confirming the existence of the clusters and characterizing its nature. These are used for visualizations of the weights associated with every neuron after training. If the attributes in the input data are highly unrelated, then this will be depicted in the heat maps of the attributes.

In Figure 29 (a) and (b) it can be observed that no relation between energy and frequency exists when training done for 500 iterations. So it can be said that network has not learned properly.

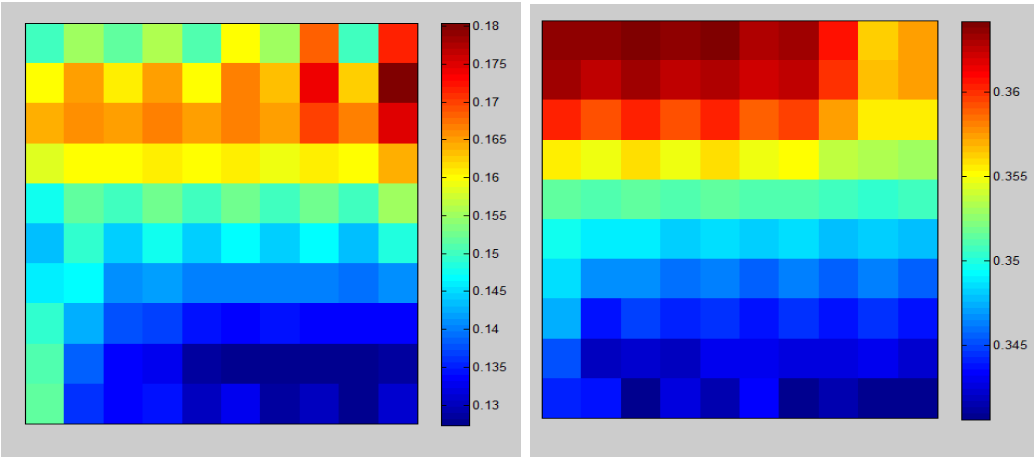


Figure 29. Heat map for 500 iterations (a) Frequency (b) Energy

In Figure 30 (a) and (b) we can observe the distribution of frequency and energy. For instance we can see that the lower left cluster is characterized by high values of frequency and energy.

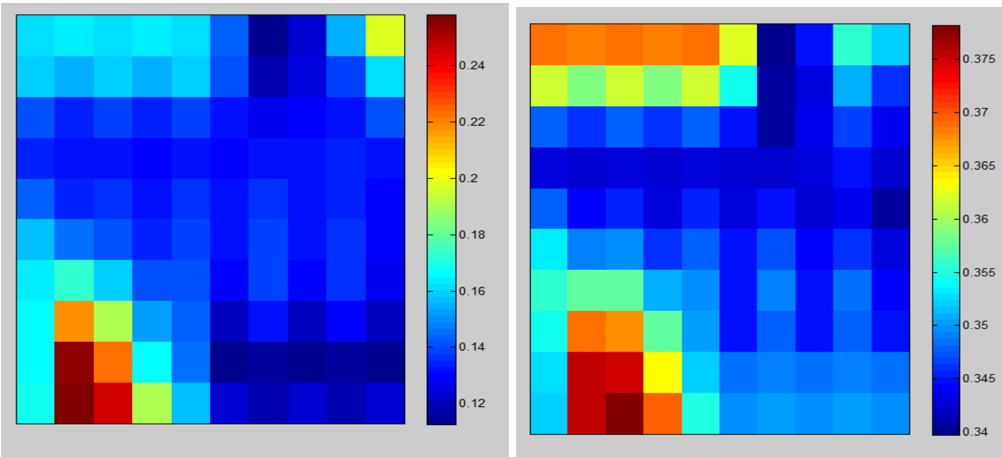


Figure 30. Heat map for 5000 iteration (a) Frequency (b) Energy

3.4.5 Seismic Image after Clustering:

3.4.5.1 Voxel Coordinates as Input:

Most likely the nearby voxels belong to the same cluster so including voxels' x, y, z coordinates as input helps in reducing topographic error, but including the voxel coordinates as input would also affect the fused image generated through the clustering of slices far apart.

It can be observed from the Figure 30 in which clustering of energy and frequency has been done for '528ms' and '1700ms' time slices. If the voxel coordinates are included as input here then the known and the unknown bright spot are not falling into the same or close by cluster.

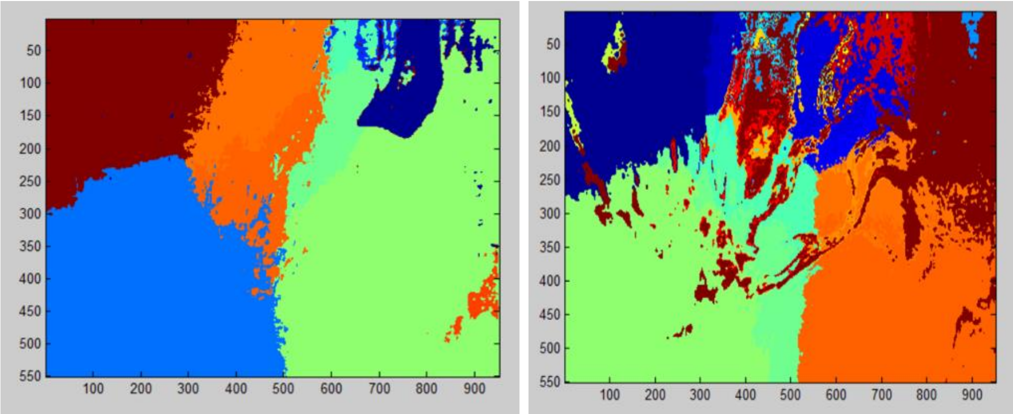


Figure 31. Clustering of energy and frequency for (a) 528ms (b) 1700ms time slices including the Distance as Input

Figure 32 shows that when voxel coordinates are not added as input known and the unknown bright spot lie in the same cluster.

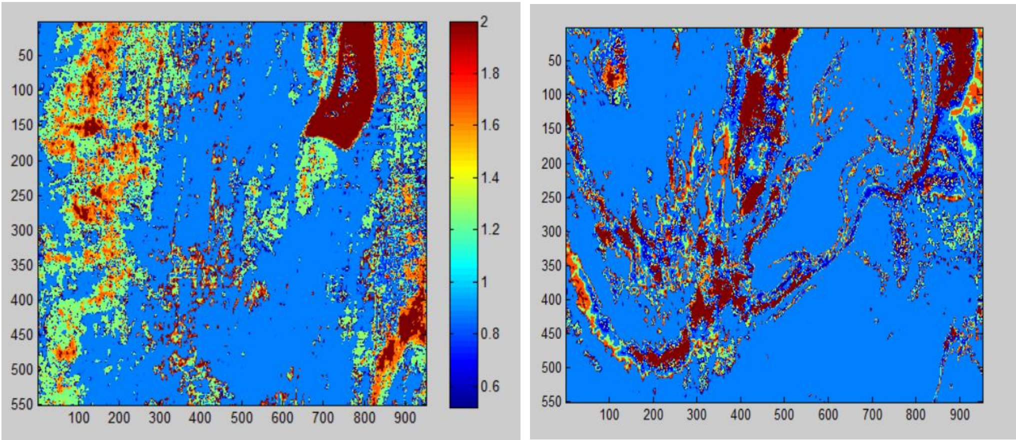


Figure 32.Clustering of energy and frequency for (a) 528ms (b) 1700ms time slices not including the Distance as Input

Figure 33 show the comaparison of clustering with and without adding the voxel coordinates for time slices ‘524ms and 528ms’. It is visible that many small clusters are formed which are not recognizable shown in the circled area.

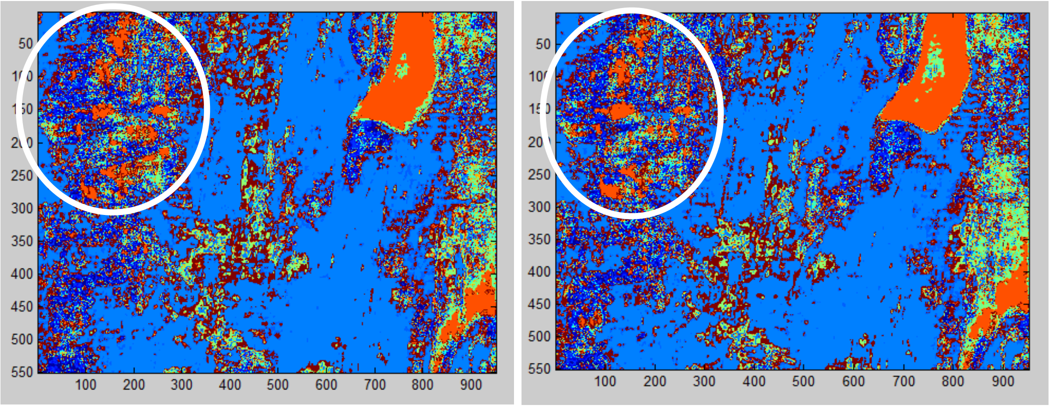


Figure 33.Clustering of energy and frequency for (a) 524ms (b) 528ms time slices not including the Distance as Input

In Figure 34 (a) and (b) more defined clusters are formed

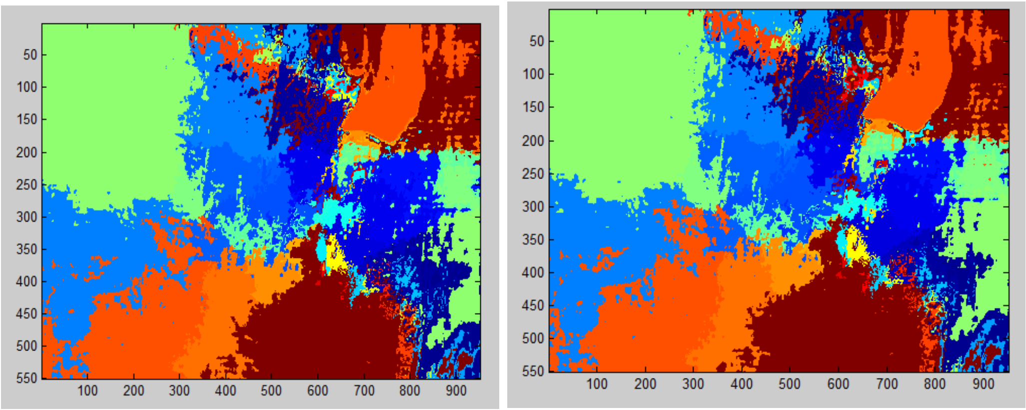


Figure 34.Clustering of energy and frequency for (a) '524ms' and (b) '528ms' time slices including the Distance as Input

So giving voxel coordinates as input is suggested only when nearby time slices are clustered.

3.4.6 Fault Detection:

For further verification of the algorithm isotropy, coherency and chaos were clustered for fault detection. In Figure 35 encircled area shows that faults are present at the 1648ms time slice. Faults are being enhanced in the Figure 36.

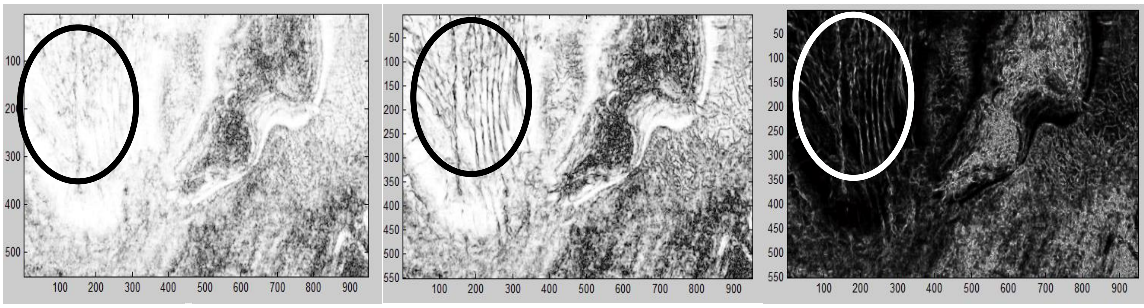


Figure 35. Image of (a) Isotropy (b) Coherency (c) Chaos for 1648ms.

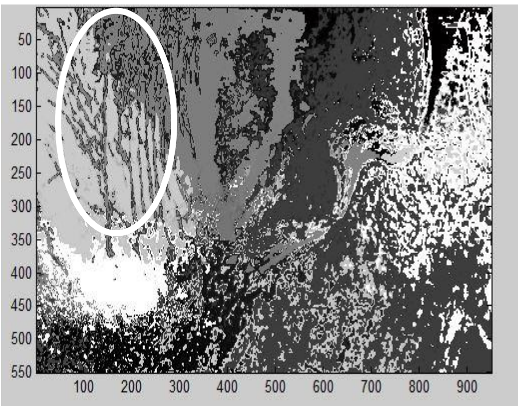


Figure 36. Clustered Image of Isotropy, coherency and chaos for 1648ms time slice

3.4.7 Optimization:

3.4.7.1 Optimum number of epochs:

The F3 block consists of 651 in lines, 951 crosslines, 462 time slices so in total input samples for training are 651x951x462= 286024662. As it is a very large number so it would definitely take ages to train. That is why the analysis has been done to find the optimum number of epochs to reduce the number of input samples used for training.

3.4.7.2 Vectorization:

The comparison in the Table 1 shows that run time is reduced after vectorization but it is still not efficient as at least 50 time slices should be clustered in minimum possible time.

Before Vectorization				After Vectorization		
Number of Slices	1	2	6	1	2	6
I3 CPU-1.7GHz	57 min	1hr 40min	7hrs 42min	6 min	45 min	3 hrs.
I7 CPU 2.7GHz	40min	1hr 12min	6hrs 32min	3 min	20 min	1hr 40min
GPU	27min	50 min	4hrs 30min	2 min	15 min	1hr 9 min

Table 1. Time Analysis for clustering of different number of Time slices (a) Before Vectorization (b) After Vectorization

3.4.7.2 MATLAB Parallel Toolbox:

MATLAB Parallel Toolbox enables the algorithm to be run on multiple cores simultaneously to increase the efficiency.

Table 2 shows the time analysis of the algorithm on different systems using MATLAB Parallel Toolbox.

Number of Slices	1	2	50
I3 CPU-1.7GHz	1min	1.5min	30 min
I7 CPU 2.7GHz	**26 sec**	**75 sec**	**20 min**

Table 2 Time Analysis for clustering of different number of Time slices using MATLB Parallel Toolbox

3.4.8 Statistical Analysis:

Quantization error and Topographic error are calculated at R. R is an integrated suite of software facilities for data manipulation, calculation and graphical display. As our data set is heavy by using parallel computing in MATLAB time efficiency is better than R.

As mentioned in the 3.2.4 section two errors can be computed for the validity of the Self-Organizing Map.

Table 3 shows the statistical analysis of the SOM.

Slices	iteration	X,Y,Z coordinates added	Quantization Error	Topographic Error
6	500	yes	0.37	0.07
6	5000	yes	0.41	0.07
2	5000	no	0.371	0.15
2	5000	yes	0.371	0.15

Table3.Statistical analysis of the SOM.

3.4.9 Opendtect Visualization:

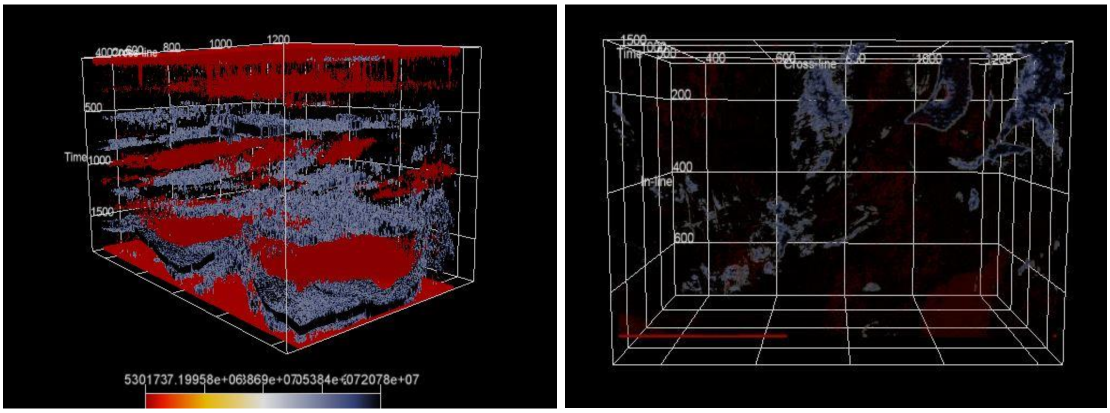


Figure 37. (a) 3D visualization of the fused image (b) Bright Spots identified at 528ms and 1700ms time slice after clustering

3.4.10 Conclusion:

It is suggested that by keeping following parameters desirable results can be achieved

i. initial learning rate as 0.1
ii. initial neighborhood radius 5 or 7 for 10x10 grid
iii. number of iterations greater than 1000
iv. Giving voxel coordinates as input is suggested only when nearby time slices are clustered.

3.4.11 Scope and Deliverables:

It will provide the geologist to easily extract features of his interest from the fused image. Then based upon his knowledge make decision for an oil digging company whether to invest their money or not.

It can be used to identify the unknown hydrocarbon regions in the given seismic survey.

Table 4 shows different attributes which can be clustered for feature detection.

ATTRIBUTES	FEATURE
Dip	Faults
Azimuth	Orientation of Faults
Energy	Bright spots indicator
Frequency	Direct hydrocarbon indicator
Isotropy	Similarity between traces direction ,fractures
Coherence	Measures Similarity between traces
Chaos	Chaotic region
Fault Enhancement	Enhance faults

Table 4. Attributes used for Feature Detection

3.4.12 Future Recommendations:

1. Attribute selection is done manually by using the background study. However it can be automatized by using Principal Component Analysis before giving input to SOM.
2. Growing SOM can be used for determining the grid size automatically.

4.BIBLIOGRAPHY:

[1]R. Roden, T. Smith and D. Sacrey, "Geologic pattern recognition from seismic attributes: Principal component analysis and self-organizing maps", *Interpretation*, vol. 3, no. 4, pp. SAE59-SAE83, 2015.

[2]A. Köhler, M. Ohrnberger and F. Scherbaum, "Unsupervised feature selection and general pattern discovery using Self-Organizing Maps for gaining insights into the nature of seismic wavefields", *Computers & Geosciences*, vol. 35, no. 9, pp. 1757-1767, 2009.

[3]T. Zhao, V. Jayaram, A. Roy and K. Marfurt, "A comparison of classification techniques for seismic facies recognition", *Interpretation*, vol. 3, no. 4, pp. SAE29-SAE58, 2015.

[4]M. Matos, K. Marfurt and P. Johann, "Seismic interpretation of self-organizing maps using 2D color displays", *Rev. Bras. Geof.*, vol. 28, no. 4, pp. 631-642, 2010.

5.REFERENCE

[1] *Tnw.tudelft.nl*, 2016. [Online]. Available: http://www.tnw.tudelft.nl/fileadmin/Faculteit/TNW/Over_de_faculteit/Afdelingen/Imaging_Science_and_Technology/Research/Research_Groups/Quantitative_Imaging/Publications/PHD_Theses/doc/thesis_pbakker.pdf. [Accessed: 13- May- 2015].

[2] M. Taner, F. Koehler and R. Sheriff, "Complex seismic trace analysis", *GEOPHYSICS*, vol. 44, no. 6, pp. 1041-1063, 1979.

[3] T. Smith, "Introduction to self-organizing maps in multi-attribute seismic data", *Geophysical Insights*, p. 10, 2011.

[4] 2016. [Online]. Available: https://opendtect.org/osr/pmwiki.php/Main/NetherlandsOffshoreF3BlockComplete4GB. [Accessed: 20- Jun- 2015].

[5] S. Chopra and K. Marfurt, *Seismic attributes for prospect identification and reservoir characterization*. Tulsa, Okla.: Society of Exploration Geophysicists, 2007.

[6] A. Iske and T. Randen, *Mathematical methods and modelling in hydrocarbon exploration and production*. Berlin: Springer, 2005.

[7] G. Manikanda.et.al, "Achieving Privacy in Data Mining Using Normalization", *Indian Journal of Science and Technology*, vol. 64, pp. 4269-4272, 2013.

[8] "Kohonen Self-Organizing Maps - Shy Pages 1 - 21 - Text Version | FlipHTML5", *Fliphtml5.com*, 2016. [Online]. Available: http://fliphtml5.com/lpyn/opkd/basic. [Accessed: 6-July- 2016].

[9] [Online]. Available: http://www.ifs.tuwien.ac.at/~poelzlbauer/publications/Poe04WDA.pdf. [Accessed: 12- Jan- 2016].

6. APPENDICES

6.1 Appendix A:

LMKR's PROJECT

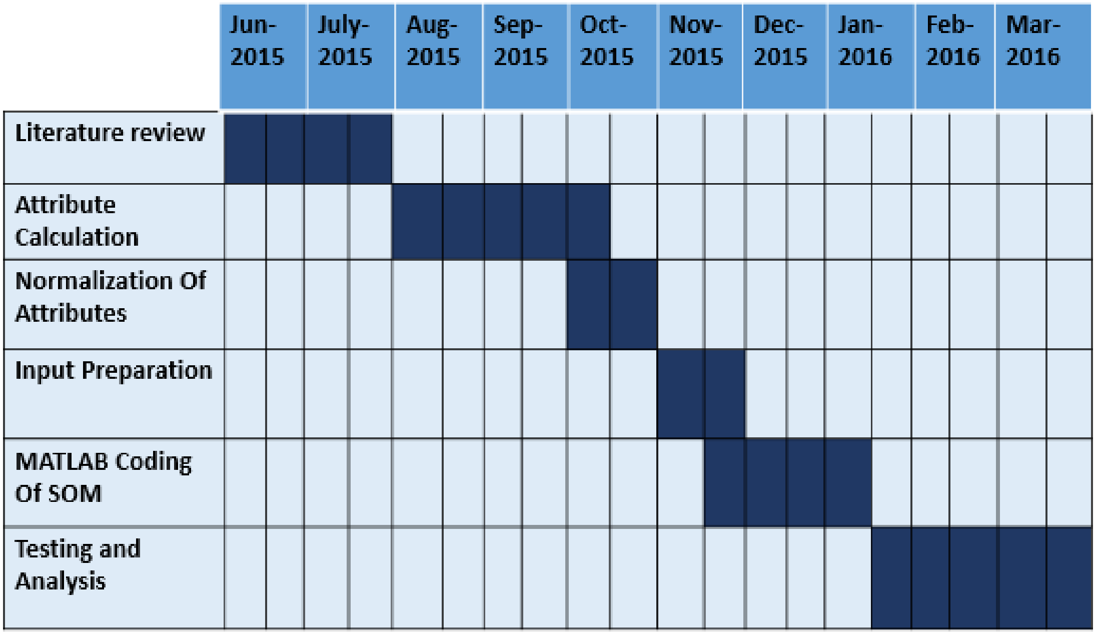
SEISMIC FEATURE DETECTION

Extended Title: Clustering of features in multiple attributes using self-organizing maps
Brief Description of The Project / Thesis with Salient Specifications: Seismic imaging directs an intense sound source into the ground to evaluate subsurface conditions and to possibly detect high concentrations of contamination. We want to design an automatic method which clusters the attribute volumes of seismic images to segment different types of features using self-organizing maps.
Scope of Work : Seismic images are large 3D volumes of the layers of the subsurface. They can be several gigabytes large. Each seismic volume can have multiple attributes. It is difficult to visualize these attributes at the same time to identify different types of features.
Academic Objectives : • Concept of artificial intelligence and statistics is being applied in the self-organizing maps. • Concept of image processing is also applied in seismic imaging and its feature detection.
Application / End Goal Objectives : A 3D volume showing the cluster to which each voxel belongs for easier visualization

Previous Work Done on The Subject : K¨ohler.A , Ohrnberger.M & Scherbaum.F (2010). Unsupervised pattern recognition in continuous seismic wavefield records using Self-Organizing Maps. Geophysical Journal International. Doi: 10.1111/j.1365-246X.2010.04709.x
Resources Required: • Perpetual interaction with LMKR for acquisition of relevant data set • Interfacing the optimized algorithm with the CPU/GPU.
No of Students Required : 4 **Group Members:** 1. **Fareeha Ikram** 2. **Zaiba Shah** 3. **Urva Latif** 4. **Muhammad Sarmad Masood**
Special Skills Required: • Learning seismic interpretation. • Optimization of algorithms.

Synopsis

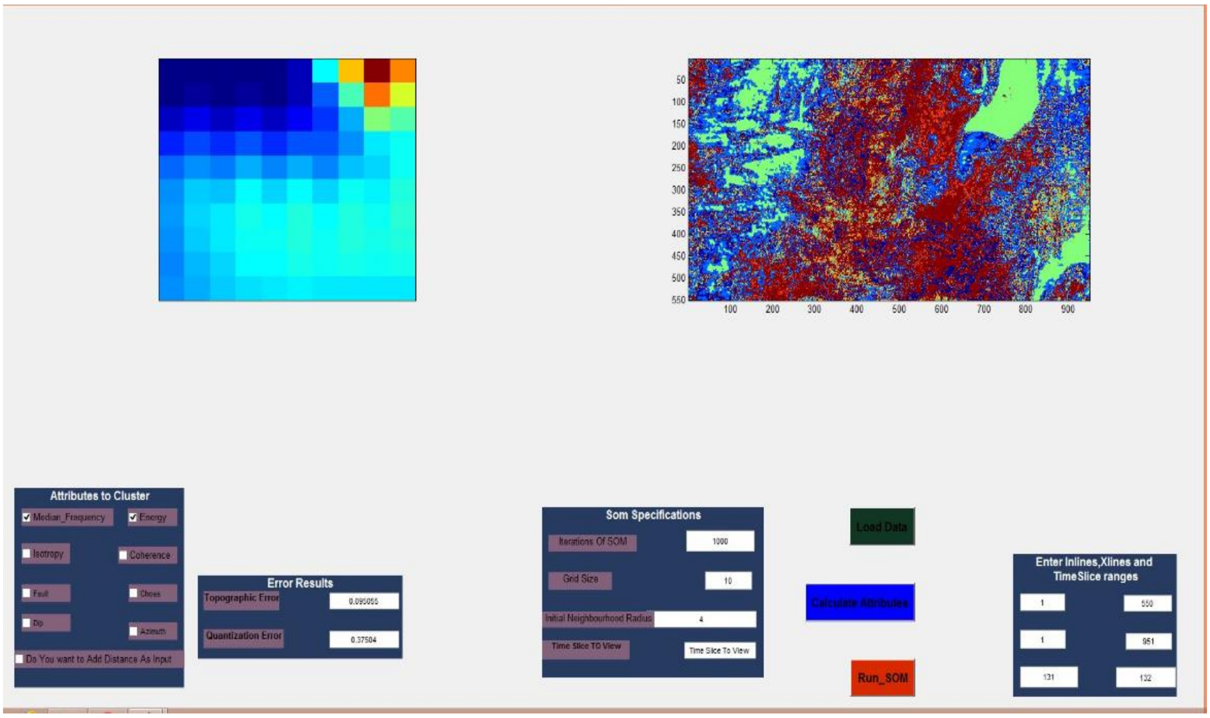
6.2 APPENDIX B



Work Plan

6.3 APPENDIX C

GUI



USER INPUT:

1. Click “Load Data” to load seismic data.

2. Click “Calculate Attribute” to calculate all the attributes.

2. Check mark the attributes to cluster.

3. Check mark the box of "Do you want to add distance as input".

4. Enter the x-line, in-line, time slices for clustering the input samples.

5. Enter the number of iteration.

6. Enter the grid size.

7. Enter the initial neighborhood radius.

8. Click Run_SOM.

OUTPUT

1. Clustered image

2.Topographic error

3.Quantization error

4. Trained weights