Boundary determination of mineral regions in hyperspectral drill core imager data

Tshepiso Mothlele

Supervisor: Professor Michael Sears

Co-supervisor: Dr. Byron Jacobs

School of Computational and Applied Mathematics,

University of the Witwatersrand,
Johannesburg, South Africa.

A dissertation submitted to the Faculty of Science, University of the Witwatersrand, in fulfillment of the requirements for the degree of Master of Science.

Declaration

I, Tshepiso Mothlele, declare that the content of the dissertation titled "**Boundary determination of mineral regions in hyperspectral drill core imager data**" is original except where due references have been made. This work is my own and work previously done by others has been stated and referenced. It is being submitted as fullfilment of the requirements for the Degree of Master of Science as the University of the Witwatersrand, Johannesburg.

Tshepiso Mothlele
October 31, 2017
University of the Witwatersrand
Johannesburg

Acknowledgments

I would like to thank my parents for their support, encouragement and patience though my academic years at university, my siblings, friends and fellow colleagues for their support and guidance though difficult and good times, my supervisor Professor Michael Sears for the guidance, support and for sharing your knowledge as I have learned a lot from you. I would also like to acknowledge the university of the Witwatersrand, the school of mathematics and the school of computational and applied mathematics for giving me so much exposure and for stretching my mind. Thanks also to my Co-supervisor, Dr Byron Jacobs from the School of Computational and Applied Mathematics, to Dr Asad Mahmood from the School of Electrical and Information Engineering for the constant support in putting my research together. Thanks to Prof. Resmini from the George Mason University for his expertise in image processing and sharing some of his work, thanks to Geospectral Imaging (Pty) Ltd for providing us with the core data for this research.

I would like to thank dear God, our heavenly Father who makes all things possible. This would not have been possible without Him.

Contents

List of Figures

Abstract

This work is about segmentation of hyperspectral images in order to identify the different regions within the image and determine their best boundaries. A new method based on the Fitzhugh-Nagumo model is introduced to do this by extracting additional useful information from the image data. It is inspired by the work done in [1] titled "A novel approach to text binarization via a diffusion-based model", which was applied on 2-dimensional images in grayscale. The results from the proposed method is then assessed and compared to other existing methods on a scene of Cuprite, Nevada, as well as drill core imager data. The datasets chosen for this research are the artificial and real AVIRIS Cuprite of Nevada and as well as real drill core imager data. Cuprite, Nevada was chosen because it is well studied and other methods have been applied to it. The core imager data is studied due to its importance of delineating minerals regions and the likelihood of getting purer pixels given the high spatial resolution of the data.

Chapter 1

Introduction

The study of hyperspectral images has been one of the most interesting among research topics in the branch of mathematical science that is image processing. It involves studying an areal view of vegetation, grounds containing different minerals, and for various other reasons with the aim of exploiting the rich information contained within these images for analysis and coming up with insights about such areas. This shows that the analysis of hyperspectral images is of great importance in industries such as mining and geology as well as in research and computing. For instance, it can help geologists to extract information from the processed hyperspectral images and have a better analysis and understanding about materials that are present in a sample that is being studied. This in turn facilitates better decision making in terms of mineral extraction. We will consider this particular example within this study.

Hyperspectral images contain hundreds of bands across the electromagnetic spectrum. It includes the visible spectrum of the red, green and blue combination of colours that we see with our naked eyes as shown in figure (1.1). In addition, they also contain the infra-red as well as the ultra-violet region and other larger parts of the electromagnetic spectrum with different wavelengths. Such an image contains a lot of information about the materials and properties from the area where the image was taken. This makes hyperspectral image data very large thus, there can be a of lot redundancy between bands and causing analysis to be tedious. The problem with having such large amounts of data is how to use it in such a way that it can be analyzed efficiently and accurately without loosing important information. This makes techniques that optimize in efficiency and accuracy to be of great important.

Each band of a hyperspectral image captures the energy level of a certain wavelength in that particular band. The energy intensities captured in the specialised hyperspectral camera are converted into data readable in processing softwares such as MatLab, which is mainly used in this research. In each of the pixels in the data will contain hundreds of values in each of the wavelength area.

In order to obtain hyperspectral images, very specialised hyperspectral cameras are used to produce these images as they have the ability to "see" beyond the visible spectrum like any other normal camera. So in order to analyse these images, dimensions of the images play an important role. On the other hand, if an image contains large pixels, i.e low spatial resolution, a lot of information in an area from where the image was taken can be represented as one value within that pixel, thus we can miss a lot about that particular area. On the contrary, if each pixel is small in size, then there is less mixing of materials captured in that pixel and more chance of finding one type of material. This means that we are more capable of accurate boundary identification, thus facilitating more better results.

The focus for this study is segmentation of hyperspectral images in order to find different regions and their boundaries present within the image. This is done on two sets of hyperspectral images namely, Cuprite image and drill core data. The Cuprite image in the Nevada desert, USA, was taken by the AVIRIS scanner [1] and the second image, drill core image which will be explained later in detail, was obtained from the company Geospectral Imaging (Pty) Ltd. These images are of interest in this research because the AVIRIS Cuprite data has been widely studied, other methods have been applied to it and it is freely available. This also makes comparison easy. On the other hand, the core data provides importance of delineating mineral regions with its high spatial resolution. It has been analysed and was provided by Geospectral Imaging (Pty) Ltd to enable this research.

An example of a drill core in figure (1.2) and (1.3) is formed by drilling out a cylinder from a rock and then cutting it again along the diameter throughout the length of this

[1] Cuprite image data is freely available on aviris.jpl.nasa.gov/html/aviris.freedata.html.

cylindrical rock. This is done in order to expose a flat surface from the cylinder, where a specialized camera will take a hyperspectral image of this surface for analysis. The hyperspectral drill core image gives us data about the different minerals and materials that are contained in the core. In order to analyze this data for the mineral regions in the image, we require some techniques to get information contained within it.

The data derived from a hyperspectral image can be considered as a three dimensional cube, in two spatial directions and the third direction is associated with the different spectra of the image. So each pixel in the image has different values spectrally across the different bands of the image.

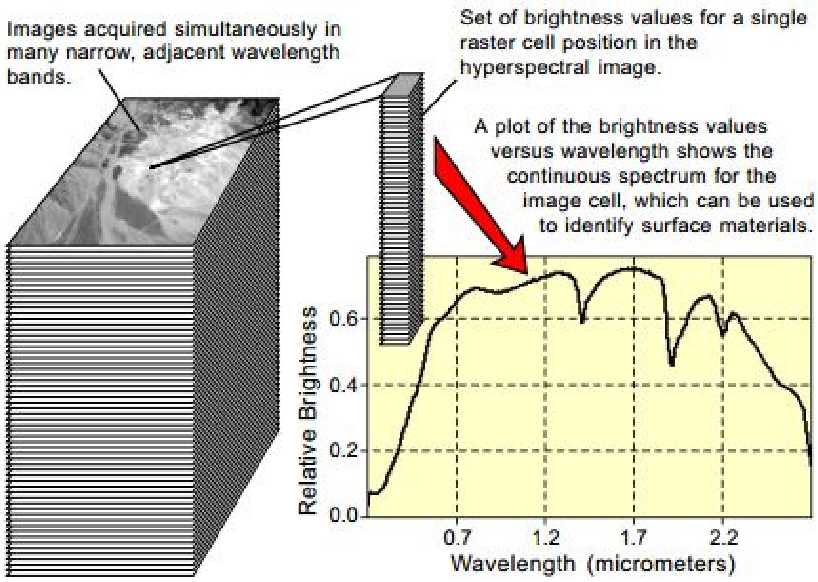


Figure 1.1: A Hyperspectral image showing different bands of an image with different energy levels at different wavelengths, taken from [2].

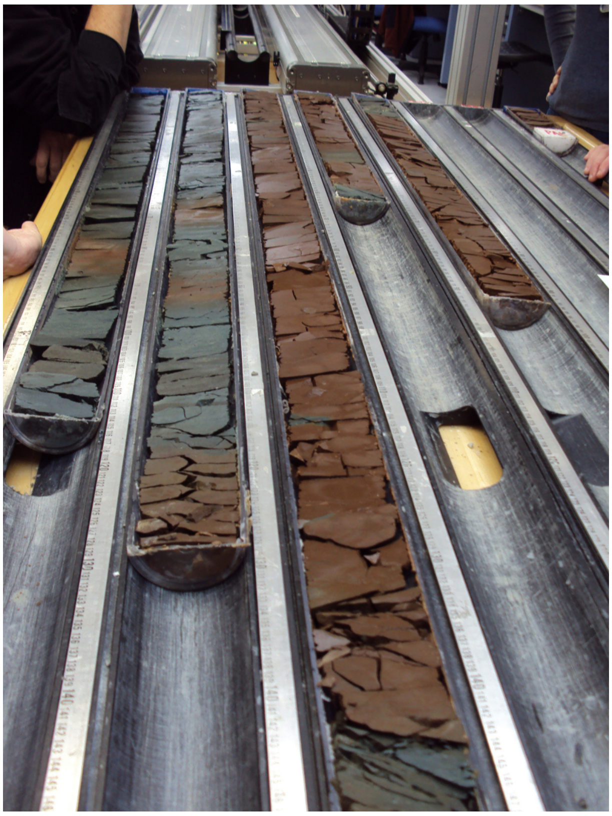


Figure 1.2: Image shows cylindrical drill cores that have already been cut in half along the length of the core (image from [3]).

Figure 1.3: Drill core image before processing (obtained from [4]).

Exploitation of the hyperspectral image and edge detection are one of the most important challenges in the study of this type of image due to their information rich nature [5]. This is because the study and research in this field helps to identify different regions and their geometrical shapes that may be contained in the sample material such as a drill core, hence in mineral exploration [2].

The artificial core is created and Cuprite image data for this research using a similar method to one used by A Mahmood, A Robin and M Sears [6]. A further explain on how the artificial data is made from the real core data in this research will be given. The purpose of introducing the artificial image is that with artificial images, we know what is contained in the data and where edges lie. When applying the proposed model, it can already be determined if the model works as it shows expected results.

The region and edge detection problems are the main purpose to be addressed by this research as well as testing our proposed method and other techniques to find these regions in the image.

This bring us to developing a new region detection method which should also provide better edges than other method for hyperspectral data when applying edge detection techniques. The method is derived from the nonlinear Fitzhugh - Nagumo diffusion equation. Our proposed model will then be compared with other methods such as the HySPADE and the SOM methods to see how it performs to give desired outcome. A very particular interest of the research is the development of the diffusion model applied to hyperspectral images. The non-linear Fitzhugh - Nagumo equation equation of the form

$$u_t = \mu\Delta^2 u + \eta f(u) \tag{1.1}$$

was explored on two dimensional gray scale images by B. Jacobs and E. Momoniat [1]. This model has a broad application in mathematics and describes, for instance, the diffusion of heat across a heat conducting plate. Now to describe the equation here in terms of heat transfer, u represents the temperature on the surface, the Laplace operator, Δ^2, has the diffusive property along the Cartesian direction and $f(u)$ is the source term of the heat, μ and η scale how effective the respective terms in the equations should be. For the purpose of this research, the scaling coefficients are varied to investigate the different effects by the model.

In the context of images, the amount of "heat" in the image pixel is determined from the pixel values of the image (where the original image is the initial condition) and the source term highlights and outlines the different pixels that are different to surrounding pixels according to a threshold value in the source term. In this research, the choice of the threshold value within the source term was based on the dynamic changes of the image via the diffusion process. It is chosen as the mean of each image band at each iteration step. However, there are different ways in which this value can be selected, one example being the Otsu's method used in [1] for the two dimensional case. We will describe later in detail the initial condition as well as the the different terms in the proposed method.

It is for these particular reasons that the diffusion model is desirable for this research. Its effects on the hyperspectral image is investigated. This is an extension to the work done by B Jacobs and E Momoniat in [1]. The 2-dimensional Fitzhugh-Nagumo model is studied and then the 3-dimensional method developed forms the crux of this research. Now with this in mind, we use this method as a technique to look for well-defined regions and edges in the data without having to analyse the data down to the endmember level. This is because the reduction of the hyperspectral scene to possibly mixed endmembers still leaves open the question of where best to draw the boundary between the different mineral facies. We hope that it will give a reasonable explanation of how the endmembers and regions are arranged for analysis. Edges contained in an image are taken to be those parts of the image where there is a sharp change of pixel values from one pixel to the next one. In a hyperspectral image there is a variety of ways of measuring when the attribute vector (spectrum) changes, and it is not easy to determine the optimal edge. This is of essence in the current application.

Chapter 2

Literature Review

Different methods of detecting edges in hyperspectral image data have been proposed, but most have a similar problem in handling the high dimensional data sets. Techniques such as Principal component analysis (PCA), Fisher's linear discriminant analysis (LDA) and unsupervised LDA applied in Matthew et. al [5] have been used to reduce the data in the image to overcome the problem of having a lot of redundant information to process. There are several techniques that have been applied in the field of hyperspectral image processing including cellular automata [5], a gradient based method [7], statistical approaches [8], the HySPADE [9] and some of the well-known techniques such as the self-organizing map approach [10] and the Canny edge detection method [11] to mention a few.

This section covers a selection of literature in the study of hyperspectral image processing for region detection. We proceed now to go through some of the research made previously.

The following were reviewed; gradient based clustering, the self-organizing map, Canny edge detection and the HySPADE approaches in detail. In addition, a nontraditional approach to the study of edges detection is reviewed, namely "A novel approach to text binarization via a diffusion-based model" [1] which is modified for the research on the 3-dimensional model. HySPADE was selected because it has a unique approach which is different to known clustering techniques as it uses spectral angles calculations. The gradient clustering approach considers the gradient of pixels across the bands, the pixel gradient magnitude and clusters the pixels into edge and non-edge pixels. And finally, we reviewed the approach that creates SOM and compares each vector of the data input to the SOM neuron vectors to convert the hyperspectral image to a gray scale image.

2.1 Gradient based clustering

The first of the three methods by R. Leitner et. al [7], is a method for edge detection based on gradient clustering. An alternative clustering of all image pixels in a feature space constructed by the spatial gradient and spectral bands is presented in the paper. It focuses on edge detection methods based on vector techniques. The aim is to use pixel gradient magnitude calculated by the "hypotenuse function" (not fully defined in the paper) and the idea is to find the direction of the point which will give the maximum rate of change for the vector in the spectral domain and determine the derivatives for different channels by Gaussian blur. Pixels are then to be clustered into edge and non-edge clusters by applying k-means method and then a method such as the Canny's detection method [11] is applied for thresholding.

For this method, each pixel is considered as a point in the spectral space having gradient values in all of the image bands, instead of intensity values. Initially, the partial derivatives of the channel in the hyperspectral image are determined using 2-D Gaussian blur convolution. The hypotenuse function is used to determine the magnitude of gradient of the pixels over all channels of the hyperspectral image. Then, to determine the edges, a clustering method is used. K-means clustering and other methods can now be used to cluster the data (determining the value of k is key for obtaining the correct number of clusters). A combined classifier method (by P. Paclik et al. [12]) is used to isolate edges and remove noise. This method combines the results in the spatial with the spectral range. The process is repeated until stable results are achieved and the results are combined using the maximum combination rule which is based on statistics (details on the method are in [12]).

To refine the results, a thresholding method is applied. The paper uses two threshold values, a higher and a lower threshold value. According to their technique, edge pixels are clustered together if the pixel value is below the lower threshold relative to the non-edge pixels. It is also an edge pixel if the value is between the two threshold values and satisfies the condition that it is connected to an edge pixel. The remaining pixels are classified as non-edge pixels. The paper also provides the algorithm of the method used (in [7]).

Based on the results obtained in the paper in comparison with the method by S. Di Zenzo [13] et al. [14], the method seems to perform well. However, it loses some of the edge information and results in discontinuous edges while the method by Zenzo gives better results. When looking at the output on the MFISH dataset [15], their method performs better which can be a potential application to the drill core data because there is not much mixing in the data and we expect to see a clear separation of regions and more accurate edges.

2.2 Self-Organising Map

One of the important methods to mention in this study is the Self-Organizing Map method (SOM) that is derived from the work of T. Kohonen [16]. The main importance of this method is for dimensionality reduction. It does this by representing higher dimensional data by lower dimensional data through mapping. It is widely used in many areas of application such as image processing [10] which is introduced in this study, also the SOM is applicable in the processing of speech shown in [16] as one example from the paper. It is also currently employed by Geospectral Imaging (Pty) Ltd as part of its processing routine. It gives the advantage of converting complex data into simpler meaningful data.

The SOM is a map to 1- or 2-dimensional set of nodes from a dataset. With these sets of data there is an associated set of vectors that form a network. These vectors are connected, meaning that similar vectors are set close to each other if they are related to the neighbouring vectors. The network is formed by the iterative SOM method, computing the next best set of vectors in the network to represent the input dataset. To describe some technical aspects, suppose that the set of SOM vectors is given by $\mathbf{m}_i$ is the **ith** node and $\mathbf{x}$ is the vector tested, and both are in $\mathbb{R}^n$, from [16] the equation

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{c(\mathbf{x}),i}(t)\left(\mathbf{x}(t) - \mathbf{m}_i(t)\right) \tag{2.1}$$

updates the vectors at each iteration t, where $h_{c(x),i}$ is a scaling parameter often taken as the Gaussian function

$$h_{c(\mathbf{x}),i}(t) = \alpha(t)exp\left(-\frac{||\mathbf{r_i} - \mathbf{r_c}||^{\mathbf{2}}}{2\sigma^2(t)}\right). \tag{2.2}$$

where $c = c(x)$ is defined by the condition

$$||\mathbf{x}(t) - \mathbf{m_c}(t)|| \leq ||\mathbf{x}(t) - \mathbf{m_i}(t)|| \tag{2.3}$$

and $m_c(t)$ is the "winning" vector which best represents $x(t)$, $\mathbf{r_i}$, $\mathbf{r_c}$ are the location vectors of the SOM vectors that define h_c, $\alpha(t)$ is such that $0 < \alpha(t) < 1$, called the learning-rate factor, and finally, σ is associated with the width of the the neighborhood function, i.e. the Gaussian function. Further description of this method and its applications is given in [10].

The second method presented by J. Parkkinen et al. [10] is based on the use of self-organizing map and grayscale edge detector. One approach in gray-level images is to see the pixel values as an ordering for pixels. The method presented in the paper is claimed to be much better than R-ordering method presented in [7] which converts a multidimensional image to gray-scale by a euclidean norm for each vector as follows,

$$d_j = \sum_{k=1}^{n} ||\mathbf{x}^j - \mathbf{x}^k||, \tag{2.4}$$

where $\mathbf{x}^j$ is the $j-th$ vector which represents the $j-th$ input pixel, d_j is a scalar that replaces the original pixel position in the gray-scale image and n is the number of pixels. Self-organizing map and Peano scans are used to convert the dataset to a gray-scale image in which edges are possible to determine with the use of Laplace and Canny operators. Stronger edges appear when there is greater deviation of gray values of the pixel from the background [10]. The method considers a multispectral image as a vector field with each pixel extended as a vector throughout the bands and will be the input for the algorithm. At each iteration stage the neuron vectors are determined which form the SOM. The input vectors are compared with the neuron vector. The following equation relates each input vector with each neuron vector and a value c [10],

$$||\mathbf{x}^j - \mathbf{m}_c^j|| = \min_i ||\mathbf{x}^j - \mathbf{m}_i^j||, \tag{2.5}$$

where $\mathbf{m^j}$ is the j-th neuron vector that form the SOM, and c is the best fitting constant which relates the vectors and the neurons to satisfy the equation. The SOM is updated at learning phase at each iteration stage. For a two dimensional SOM, a Peano scan (fractal curve in B. Mandelbrot 1977 [17], the properties in R. Stevens et al. [18], 1993) is applied and 2-dimensional matrix $\mathbf{P} = [\mathbf{p^1}, ..., \mathbf{p^k}]$ orders the vectors, where $\mathbf{p^i}$ is a Peano vector for k number of neurons described further in [10]. Peano scan is used to get the ordering of the image vector for the paper; the authors use it for mapping a set of points in n-dimensions to give a 1-dimensional image. In the paper, the Peano scan is quantized to match the size of the SOM. The number of the columns of P is the same as the number of neuron vectors in the SOM. So, now each vector in the original image is matched to each column of P by the scalar c in $\{1, 2, ..., k\}$, where k is the number of columns of P. The relationship is as follows,

$$||\mathbf{x} - \mathbf{p}^c|| = \min_i ||\mathbf{x} - \mathbf{p}^i||. \tag{2.6}$$

The scalar values are used to create a gray scale image. In the case where the SOM is 1-dimensional the neighbourhood N_c is defined as

$$N_c = \{max(1, c-1), min(k, c+1)\}. \tag{2.7}$$

This is the neighbourhood centered on the neuron vector for which the best match with input $\mathbf{x}$ is found [10]. The R-ordering method is less effective to determine edges as compared to the method presented in [7]. This is because edges may be "erroneous" in the sense that they do not actually represent the information in the image . This can be corrected by the method [10]. It is also emphasized in the paper that the dimensions of the SOM do not matter for the final results, but what matters is the size of the SOM that is to be taught. So, if the SOM is too small, the output of the image showing edges will be poor as compared to a better image from teaching a larger SOM. Clustering methods may have a disadvantage that at different bands or channels, the derivatives may have to be scaled differently in order to obtain good results [8]. The SOM technique is currently being used by Geospectral Imaging (Pty) Ltd as part of its processing routine.

The figures below are an example of results obtained using the SOM method and K - means clustering from the report by P. Padayachee [19].

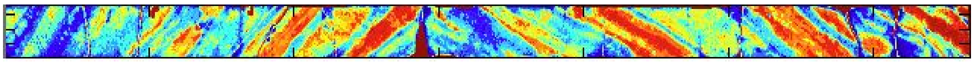


Figure 2.1: Drill core image after processing with 1D SOM. The results were obtained from [19] for figure 1.3 as input.

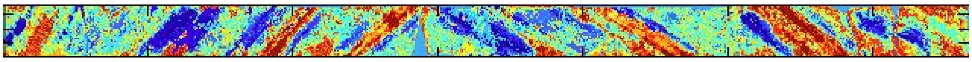


Figure 2.2: Drill core image after processing with 2D SOM. The results were obtained from [19] for figure (1.3) as input.

The author's results suggests that 1-dimensional SOM performs better than the the 2-dimensional and thus there is no need to have additional dimensions.

2.3 Canny edge detection

In the paper [11], development of the mathematical formulation for Canny edge detection is developed based on three criteria. The three criteria include low error rate, and that the edge point will be localized meaning that the distance between the points marked by the detector and where the actual edge lies is minimal. The third criterion is to circumvent the possibility of multiple responses to a single edge. In order to develop the analysis, a one dimensional edge profile was considered on the assumption that the two dimensional edges locally have a constant cross section in some direction. It was also assumed that the image has edges and white Gaussian noise. The formulation of the detection problem is formulated by:

- Beginning with a known one dimensional edge with Gaussian noise and convolved with a filter
- marking the centre of the edge at the local maximum of the convolution
- the problem becomes one to find a filter that gives best results

The criteria needed for the detected edges to be as close as possible to the real edge is formulated mathematically by the response convolution integral at the centre of the edge given by

$$H_G = \int_{-W}^{W} G(-x)f(x)dx, \tag{2.8}$$

where, $f(x)$ is the impulse response of the filter, and $G(x)$ is the edge and $x = 0$ is the centre of the edge and assuming that the impulse response is bounded by $[-W, W]$. The first criterion is thus defined by

$$SNR = \frac{\int_{-W}^{W} G(-x)f(x)dx}{\eta_0\sqrt{\int_{-W}^{W} f^2(x)dx}} \tag{2.9}$$

which is the output signal-to-noise ratio, the ratio of convolution response and root-mean-squared response to the noise n(x) only is given by

$$H_n = n_0 \left[\int_{-W}^{W} f^2(x)dx\right]^{\frac{1}{2}}. \tag{2.10}$$

The second criterion, which is the localization criterion derived by Canny [11], is defined as by

$$Localization = \frac{\left|\int_{-W}^{W} G(-x)f'(x)dx\right|}{\eta_0\sqrt{\int_{-W}^{W} f^2(x)dx}}. \tag{2.11}$$

This can then be formulated as a product of the localization and SNR given by

$$\frac{\left|\int_{-W}^{W} G(-x)f'(x)dx\right|}{\eta_0\sqrt{\int_{-W}^{W} f^2(x)dx}} \frac{\left|\int_{-W}^{W} G(-x)f(x)dx\right|}{\eta_0\sqrt{\int_{-W}^{W} f'^2(x)dx}}. \tag{2.12}$$

It measures the effectiveness of the filter to discriminate between signal and noise at the centre and edge and does not take into consideration how the filter behaves nearby the edge centre. The two criteria become a maximisation problem and by using the Schwartz inequality, it can be shown from [11] that the SNR is bounded above by

$$\eta_0^{-1} \int_{-W}^{W} G^2(x)dx, \tag{2.13}$$

and the localization given by

$$\eta_0^{-1} \int_{-W}^{W} G'^2(x)dx. \tag{2.14}$$

The localisation and the SNR are both maximised in $[-W, W]$ when $f(x) = G(-x)$. Now, when $G(x)$ is the step edge, defined as

$$G(x) = Au_{-1}(x), \tag{2.15}$$

where $u_n(x)$ is the nth derivative of the delta function, and A is the amplitude of the step. We have

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

Now substituting for G in the SNR and localization gives

$$\frac{A\,|f'(0)|}{\eta_0\sqrt{\int_{-W}^{W} f^2(x)dx}}. \tag{2.16}$$

Therefore, by substitution

$$\sum f \wedge f' = \frac{\left|\int_{-W}^{0} f(x)dx\right| |f'(0)|}{\sqrt{\int_{-W}^{W} f^2(x)dx}\sqrt{\int_{-W}^{W} f'^2(x)dx}} \tag{2.17}$$

where $\sum$ and $\wedge$ are the performance measures of the SNR and the localization, respectively. The maximization problem becomes an expression (2.17), which has a solution which is a class of functions related by special scaling. This (2.17) is called the optimal filter.

The problem is then transformed into a constrained maximisation that involves only an integral functional to apply calculus of variations. Using Lagrange multipliers for constrained function, the constrained optimisation can be reduced to an unconstrained optimisation which becomes a minimisation problem of the integral in the denominator of the SNR term such that other integrals remain constant.

$$\int_{-W}^{0} f(x)dx \tag{2.18}$$

such that

$$\int_{-W}^{0} f(x)dx = c_1, \int_{-W}^{0} f'^2(x)dx = c_2$$
$$\int_{-W}^{0} f''^2(x)dx = c_3, f(0)dx = c_4.$$

To solve this, seek a functional form $\int_a^b F(x, f, f', f'')$ with constraints $\int_a^b G(x, f, f', f'') = c_i$. This functional is a linear combination of the function that appears in the constraints. Finding the unconstrained maximisation problem is equivalent to solving the unconstrained problem. An Euler equation that corresponds to the composite function was then determined to be of the form

$$\Phi_f = \frac{d}{dx}\Phi_{f'} + \frac{d^2}{dx^2}\Phi_{f''} = 0. \tag{2.19}$$

This equation is then solved by substituting $\Phi = \int_a^b F(x, f, f', f'')$ into (2.19), which after a series of mathematical formulations and manipulations described in detail in [11] results in the solution

$$f(x) = a_1 e^{\alpha x} sin(\omega x) + a_2 e^{\alpha x} cos(\omega x) + a_3 e^{-\alpha x} sin(\omega x) + a_1 e^{-\alpha x} cos(\omega x) \tag{2.20}$$

such that

$$f(0) = 0, f(-W) = 0, f'(0) = s, f'(-W) = 0 \tag{2.21}$$

with s an unknown constant, the slope of the function f at the origin. Now, $f(x)$ is symmetric with $f(-x) = f(x)$ meaning the domain becomes $[-W, W]$. Using the boundary conditions, the $a_i, i = 1, 2, 3$ coefficients can be determined in terms of α, ω, c and s. The solution function is then parameterized using the remaining coefficients. By determining the maximum of f at $x = 0$, a Gaussian random variable with an associated probability p_f such that for $r \approx 1$ representing the quality of a filter such that

$$r \sum = \frac{|f'(0)|}{\sigma_s}. \tag{2.22}$$

Numerical methods were employed to determine the value of r and approximate the filter using the Gaussian first derivative described in [11]. Canny also further estimates noise and applying thresholding to the 1-dimensional and 2-dimensional problems as well as exploring the operator width for optimal detection.

2.4 HySPADE algorithm

The HySPADE approach [9] uses the spatial and also spectral information in order to determine edges (originally published in [20]). An edge detector is then applied using a matrix window to the spectrum obtained. A thresholding algorithm is then applied. As an outline of the HySPADE algorithm, the hyperspectral image (HSI) data cube is taken in as input (as the initial step of the algorithm) calibrated for example as radiance spectra. For any user-selected n, smaller than the number of pixels in any spatial direction, an nxn sliding window is used to process the nxn pixels within the window on the data. The window is advanced to the next set of pixels for processing.

The user chooses by how many pixels the window moves (according to the paper, the best way is to slide by $n-2$ pixels). Now spectral angle mapper (SAM) algorithm is applied on each pixel in the window with each and every other pixel in the window to construct the spectral angle (SA) cube. The cube becomes nxnxn^2 in size. Each spectrum in the SA cube has spatial position information about material in the window.

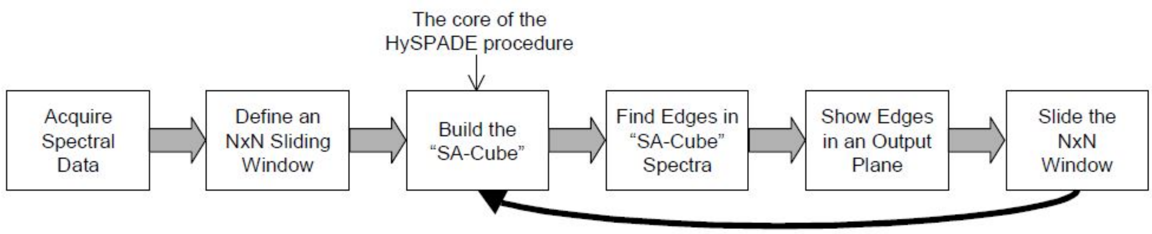


Figure 2.3: An overview of the HySPADE method (obtained from [9]).

In analysis of the SA cube, each spectrum of the cube is analyzed in turn to detect edges by a one-dimensional detection technique (using finite-difference in the HySPADE). The finite difference method is applied through the spectral angle values obtained for a pixel which produces different values. A value that exceeds a selected threshold is considered as an edge. So, when an edge is detected in the SA cube spectrum, a point is plotted separately as output for the edge (the band number of the cube is translated into the [sample, line] address/position in the original window). The entire process in applied twice, first with the window sliding in the row direction and then in the column-wise direction. So the HySPADE records pixel by pixel the number of times a given [sample, line] address indicates an edge from the edge detection (in 1 dimension). The edges are determined from values that exceed the user-defined threshold.

It is seen from the paper that the threshold differs for each spectrum processed. The standard deviation σ of the first-order finite-difference edge detection values calculated from each SA cube spectrum is used as the threshold value. By creating a set of threshold values by multiples of σ, $\{0.2\sigma, 0.4\sigma, ..., 4\sigma\}$, different output planes of detected edges can result. The output can be enhanced using techniques such as histogram stretching. Instead of using spectral angle, one can use the Euclidean distance measure which will put emphasis on spectral and brightness differences while spectral angle emphasizes spectral differences without regard for illumination or overall brightness differences among pixels of the same material type. The method's performance depends on the type of data that is initially used.

2.5 Sobel, Prewitt and Robert's operators

Other types of edge detection techniques that are commonly applied, among others such as Canny edge in the field of edge detection, is the Sobel and Prewitt methods. The two techniques are similar except for the factor of 2 used in Sobel mask in their 3x3 convolution matrices [21]. The reason for introducing these techniques is so we can compare the results when the hyperspactral data is converted into a 2-dimensional image. This conversion can be done in a similar fashion as in [20] by summing up a number of bands of the image being analysed or by simply taking the principal components of the image. This way we will see how the proposed method compares to the commonly used Sobel and Prewitt techniques. Like the Canny edge detection technique, the two packages are readily available as part of MatLab image processing tools.

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}. \tag{2.23}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}. \tag{2.24}$$

Equation (2.23) is the Prewitt and (2.24) is the Sobel mask that approximates the derivatives in the respective x- and y-direction. The masks are convolved with the entire image as a moving window on the entire image to produce edges when either the x- or y-direction has the greatest magnitude. The convolution of the kernels and the image computes the vertical and the horizontal gradients according to G_x and G_y which are then used to compute the edge magnitude which is given by

$$|G| = \sqrt{G_x^2 + G_y^2}. \tag{2.25}$$

Robert's operator is similar to the previous described operators except that it is a two dimensional diagonal matrix. The operators are described as

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{2.26}$$

2.6 Fitzhugh - Nagumo nonlinear diffusion equation

Binarization of image pixels is another technique that has been implemented to reveal objects or features to stand out from the background. Among many other applications of the diffusion equation such as simulation of heat transfer on a plate surface, it is also applied on a two dimensional gray-scale image. This is a method that has been applied by B Jacobs in [1] that creates a binary 2-dimensional image. The process is enabled by the Fitzhugh-Nagumo equation, given by

$$\frac{\partial u}{\partial t} = \mu \left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y} \right) + \eta u(1-u)(u-\alpha), \tag{2.27}$$

$$u(x, y, 0) = u_0 \tag{2.28}$$

$$u_x(0, y, t) = 0, u_x(x_N, y, t) = 0 \tag{2.29}$$

$$u_y(x,0,t) = 0, u_y(x,y_N,t) = 0, \tag{2.30}$$

where, $u = u(x,y,t)$ is the pixel value in the (x,y) plane position at time t, α is the threshold, μ and η are the respective diffusion and source term scaling factors, and the initial condition u_0 is the original input image before processing. $x \in [0, x_N]$, $y \in [0, y_N]$ and $z \in [0, z_N]$, so x, y and z take values from 0 to x_N, y_N and z_N respectively at the boundaries. The partial derivatives blur the image while the product (source) term drives the binary effect on the image.

The Fitzhugh-Nagumo diffusion equation will sometimes be referred to as the "FN" model and details on describing this equation will be given in chapter 4.

2.7 Multivariate statistical and Cellula automata techniques

Now to mention briefly two other image processing techniques introduced in [8] and [5]. The statistical method by S. Verzakov et. al. [8]. It makes use of probability density functions to relate neighboring pixels and determine if a pixel forms an edge. Another concept for pixels that occur less often and do not form edges called the Conditional probability density function was also considered. This is because this method takes an approach that edges are a rare occurrence in the entire image [8] and this is not a natural model for hyperspectral drill core in that it may select non-edge pixels as edges due to their assumption. Another method, to briefly mention, is the cellular automata which has two parts to it. The first part makes use of standard edge detection algorithm image and adaptive threshold such as the Canny edge detection. The second part helps refine edges by a method designed to imitate biological cell interactions [5].

2.8 Mumford-Shah technique of Boundary determination by minimizing functional

The technique proposed by D. Mumford and J. Shah [22] finds boundaries by determining a pseudo-minimum on a suitable chosen function. It is done by solving Euler-Lagrange equations for the boundaries under assumptions that the boundaries are not too curved or then they use “hill climbing”. In their paper, they provide an example where they segment an interval into sub-intervals where a one dimensional signal is roughly constant.

The authors study an approach to locate boundaries in noisy signals. The following is a description on the ideas used to develop the technique. Given an input signal $g : \mathbb{S} \rightarrow \mathbb{R}$ where; $\mathbb{S}$ is is either an open region in Euclidean $\mathbb{R}^d$ or in the set interval vectors $\mathbb{L} \cap \mathbb{R}$ in $\mathbb{R}$, $\mathbb{R}$ being the whole space and $\mathbb{L}$ is lattice of integral vectors. What g does is that it converts pixels from $\mathbb{S}$ into an image in $\mathbb{R}$. This then proceeds to determine the ideal smoothed signal $f : \mathbb{S} \rightarrow \mathbb{R}$ and a set of boundaries $\mathbb{B} \subset \mathbb{R}$. Both f and g may have values on any real number in $\mathbb{R}$ or restricted to ± 1. We note that if $\mathbb{S} = \mathbb{R}$ then f is differentiable on $\mathbb{S} - \mathbb{R}$ but may be discontinuous across $\mathbb{B}$. The authors also notes the following;

- if $\mathbb{S} = \mathbb{R}$ then $\mathbb{B}$ is a codimension 1, piecewise smooth hypersurface in $\mathbb{R}$
- if $\mathbb{S} \cap \mathbb{R}$ then $\mathbb{B}$ is a union of a small rectangular hyperplane pieces parallel to the coordinates and halfway between the lattice points.

f is the reconstructed clean signal and $\mathbb{B}$ is the set of all the edges on the image, on application to image processing.
Minimization of the energy functional E is required in order to construct f and $\mathbb{B}$. If $\mathbb{S} = \mathbb{R}$, then E is defined as

$$E(f, \mathbb{B}) = \mu^2 \int_{\mathbb{R}} (f - g)^2 + \int_{\mathbb{R}-\mathbb{B}} \parallel \triangledown f \parallel^2 + \mu\eta \text{ (volume of } \mathbb{B}) \tag{2.31}$$

If f consists of integral vectors in $\mathbb{R}$, then

$$E(f, \mathbb{B}) = \mu^2 \sum_{i \in S} (f_i - g_i)^2 + \sum_{i' \in \mathbb{S}} (f_i - f_{i'})^2 + \mu\eta \text{ (volume of } \mathbb{B}) \tag{2.32}$$

with i, i' adjacent, $\bar{ii'}$ not separated by $\mathbb{B}$, where μ and η are parameters such that $\frac{1}{\mu}$ has dimensions of a distance and η has same dimension as f^2. If f has the values ± 1 and assume that $\mathbb{B}$ is empty, then E becomes

$$E(f, \phi) = -2\mu^2 \sum_{i \in S} f_i g_i + 4(\text{volume of } \mathbb{B}_f) + C' \tag{2.33}$$

for the boundary $\mathbb{B}_f$ consisting of all hyperplane patches seperated by adjecent vectors i, i' with $f_i \neq f_{i'}$. C depends only on f and $\mathbb{S}$. The authors then minimize for f, $\mathbb{B}$ by solving for $\mathbb{B}$ for μ very large, secondly eliminating parts of $\mathbb{B}$ corresponding to local extrema of E which are not local minima and thirdly, using straight "hill-climbing" to solve for nearly local minima. This is by small changes until a local minimum of E is achieved. In the proceeding, the authors worked on a variation problem for the one dimensional case where $g(x)$ is a continuous function on $a \leq x \leq b$ and $f(x)$ continuously differentiable on each interval $a_i \leq x \leq a_{i+1}$ such that $a = a_0 < a_1 < ...a_k < a_{k+1} = b$ for all k. Now, let

$$E(f, \{a_i\}) = \mu^2 \int_a^b (f - g)^2 dx + \sum_{i=0}^{k} \int_{a_i}^{a_{i+1}} \left(\frac{df}{dx}\right)^2 dx + \mu\eta k \tag{2.34}$$

for fixed parameters η, μ. Thus

$$0 = \frac{1}{2}\delta E = \mu^2 \int_a^b (f - g)\delta f dx - \sum_{i=0}^{k} \int_{a_i}^{a_{i+1}} f' \delta f dx \tag{2.35}$$

for the first variation of E by assuming that f, a_i minimizes E and then fixing a_i and replacing f by $f + \delta f$ where $\delta f(a_i) = 0$. The authors also notes that

$$f' = \mu^2 (f - g) \tag{2.36}$$

on each interval and that

$$\delta f(a_i^{\pm}) f'(a_i^{\pm}) = 0 \tag{2.37}$$

if δf has limits $\delta f(a_i^-)$ and $\delta f(a_i^+)$ as $x \to a_i$, hence

$$f'(a^{\pm}) = 0. \tag{2.38}$$

(2.36) and (2.38) are an elliptic boundary value problem. The authors describe the Green's function for (2.36) as

$$K_\mu(x) = \frac{\mu}{2} e^{-\mu|x|} \tag{2.39}$$

and hence

$$f_\mu(x) = \int_a^b K_\mu(x-y) g(y) dy \tag{2.40}$$

solves to (2.36) in the interval $[a, b]$. Then the solution to (2.36) and (2.38) on each interval $[a_i, a_{i+1}]$ is

$$f(x) = f_\mu(x) + C_{1,i} K_\mu(x - a_i) + C_{2,i} K_\mu(a_{i+1} - x) \tag{2.41}$$

where

$$C_{1,i} = \frac{2}{\mu^2(1-\alpha_i^2)}(f'_\mu(a_i) - \alpha f'_\mu(a_i + 1)) \tag{2.42}$$

$$C_{2,i} = \frac{2}{\mu^2(1-\alpha_i^2)}(f'_\mu(a_{i+1}) - \alpha f'_\mu(a_i)) \tag{2.43}$$

such that $\alpha_i = e^{-\mu(a_{i+1}-a_i)}$. Next, a_i is varied and f is differentiable on each $a_i + \delta a_i$. It is the determined from (2.38) that the small change in the energy function becomes

$$\begin{aligned} \delta E(f, \{a_i\}) &= \int_{a_i}^{a_i + \delta a_i} (f_{new} - g)^2 - (f_{old} - g)^2 \\ &= [(f(a_i^+) - g(a_i))^2 - (f(a_i^- - g(a_i))^2]\delta a_i + O(\delta a_i^2) \end{aligned} \tag{2.44}$$

which gives the following if $f(a_i^+) = f(a_i^-)$

$$[(f(a_i^+) - f(a_i^-)][f(a_i^+ + f(a_i^-) - 2g(a_i)] = 0. \tag{2.45}$$

The value of E is reduced by omitting the break point a_i to get

$$\frac{1}{2}[f(a_i^+) + f(a_i^-)] = g(a_i). \tag{2.46}$$

By substitution, Equations (2.41) and (2.42) then give

$$\frac{1}{2}[f(a_i^+) + f(a_i^-)] - g(a_i) = \frac{1}{\mu} f'_\mu(a_i) + \frac{1}{\mu} O(a_i, a_{i+1}) sup \left| f'_\mu(x) \right| \tag{2.47}$$

thus so $e^{\mu(a_{i+1}-a_i)} \to 0$ as $\mu \to \infty$ and so $f'_\mu(a_i) \sim 0$. So for μ sufficiently large, boundaries defined by minimizing E will be zero of the second derivative of the smooth f_μ of g. The paper proceeds to numerically determine the minima of E by making use of the previous workings on f through numerical integration. It is noted that $f_\mu(x)$ does not depend on the breaking points a_i. It is then proceeded as followings: let

$$f_{\mu-}(x) = 2\int_a^x K_\mu(x-y)g(y)dy \tag{2.48}$$

$$f_{\mu+}(x) = 2\int_x^b K_\mu(x-y)g(y)dy \tag{2.49}$$

and thus

$$f_\mu(x) = \frac{1}{2}[f_{\mu+} + f_{\mu-}] \tag{2.50}$$

$$f'_\mu(x) = \frac{\mu}{2}[f_{\mu+} - f_{\mu-}] \tag{2.51}$$

with $f_{\mu-}(a) = 0 = f_{\mu+}(b)$ while $f_{\mu-}(a)$ and $f_{\mu+}(b)$ satisfy $f'_{\mu-}(a) = \mu[g(x) - f_{\mu-}(x)]$ and $f'_{\mu+}(b) = -\mu[g(x) - f_{\mu+}(x)]$, respectively these can be solved numerically as will be highlighted. It is done by subdividing the interval $[a,b]$ into n equal subintervals where $h = \frac{b-a}{n}$ and $x_j = a + jh$ where $0 =\leq j \leq n$. In the paper, each a_j is assumed to be a point that occurs between the lattice points and treated as entire open interval between two adjacent lattice points. The breaks occur between lattice pairs $\{x_j, x_{j+1}\}, ..., \{x_k, x_{k+1}\}$ with defined integers $0 < j_1 < ... < j_k < n$. Now for each of the intervals $[x_{j_{i+1}}, x_{j_{i+1}}]$, the following algorithm is applied:

- calculate α_i, $C_{1,i}$, $C_{2,i}$ for $a_i = x_{j_i+1}$ and $a_{i+1} = x_{j_{i+1}}$ in (2.41) and (2.42).
- use trapezoidal rule to restrict the value of E to the interval, then the following procedure is used to solve minimizing breaks
- choose initial breaks, usually chosen from zero-crossings of $f_\mu(x)$ or which are same, zero-crossings of $f_\mu - g$.
- to calculate j_i's at each iteration, the value of j_i is varied in turn while holding the others fixed at their current value determined from previous iteration. Choose a value for j_i which gives maximal reduction on the functional E and fixing the maximum allowed change in integers $\{j_i\}$.

In summary, the examples and results are further analyzed in the paper [22]. D. Mumford and J. Shah developed a method of segmenting an image into regions by using a minimization functional in order to determine boundaries. This is derived from the length of the hypothetical boundary, the gradient of the actual smoothed image as well as difference of the true and smoothed image [22].

2.9 Conclusion

Among the methods mentioned in the literature review, the HySPADE, the Fitzhugh-Naghumo approach and also Sobel and Prewitt as standard edge detection techniques are chosen for this research. The HySPADE approach is ideal as it gives a different approach to edge detection because there are no clustering methods incorporated and we will be able to compare its results with the SOM clustering method. Previously, clustering methods such as SOM have been applied on the core imager data provided by Geospectral Imaging (Pty) Ltd.

The three main papers presented here consider the data to be analyzed as a vector space and treat each pixel as a vector in the space. This enables the exploitation of spatial and spectral information for analysis. Yet another common issue in each of the papers is obtaining the overall result from the entire data i.e. the process of combining the spatial and the spectral features of the data to produce a two dimensional image revealing the edges.

It is seen that the first two methods (R. Leitner [7] and J. Ansamaki [10]) use a clustering approach to the data while the HySPADE approach [9] is different to these in that it uses spectral angle calculations. Because the drill core problem with its high spatial resolution is rather different from most hyperspectral remote sensing applications, we want to try different techniques to using Self-organizing map and other clustering methods to detect edges in the drill core and be able to compare the results. Thus, the HySPADE and the proposed 3-dimensional fitzhugh-Nagumo nonlinear diffusion equation are suitable candidates for this study.

Chapter 3

Methodology

In this chapter we go through the actual methods used to prepare data, present and go through the nonlinear diffusion equation.

As mentioned in the literature review, the HySPADE method make use of spectral angles of each image pixel to determine where edges lie in the image. In addition, HySPADE has been applied on the Cuprite of Nevada hyperspectral image presented in [23], to which we can compare both real and artificial images. The Fitzhugh-Nagumo approach applied directly to the input data, derived from the hyperspectral image, helps us to be able to process the 3-dimensional image with the spectral signature of the different minerals that may be found. This is because of its diffusion and source term property. This development is motivated by the Fitzhugh - Nagumo model on 2 - dimensional images presented by B. Jacobs in [1].

Now we go on to describe the data to be used.

These methods are applied first on an artificial dataset where we know about the information that is contained within the data in order to be sure that the methods work properly. The methods are then to be applied to real datasets of the drill core image and assessed by experts in hyperspectral core imaging. The nature of the core images is that the minerals in the image do not have much mixing. The pixel size of the image is $1mm$ by $1mm$ and that contributes to the desired separation of individual minerals facies in the image pixels. Hence, this will facilitate the accurate processing and highlighting of the individual minerals by applying the Fitzhugh-Nagumo model. So, for both methods, the core data is much less “busy” as compared to the Cuprite image, so we therefore expect good results in terms of separation.

In this chapter, we further explain how the artificial data is prepared and go through the development of the 1-dimensional and 2-dimensional Fitzhugh-Nagumo model.
The artificial test datasets used in this research have been produced using MatLab. These types of images are based on the real images and are done mainly to remove noise. The first is an artificial dataset based on the Nevada Cuprite hyperspectral image acquired by the AVIRIS scanner. This dataset was introduced in [24]. Some bands of the image with disturbance such as atmospheric noise were removed as done by A. Mahmood et. al. [6]. In order to create the dataset, a real hyperspectral image was used to extract the endmembers and then proceeding to remixing the abundances with an appropriate set of artificial endmembers.

The following steps are used to produce the artificial image datasets:

- **Step one:** estimate the noise covariance matrix in the image, as described in [6].
- **Step two:** estimate the appropriate number of endmembers to explain the image as described in the paper “Using random matrix theory to determine the number of endmembers in a hyperspectral image” [25].
- **Step three:** determine the appropriate set of endmembers. (These will not be pure materials in general.)
- **Step four:** Remixing the set of selected endmembers, using the abundance matrix previously determined, to obtain a new image without random noise.
- **Step five:** add Gaussian noise to the resulting image.

This gives us a completely controllable and predictable dataset where we can examine the effects of the edge detection algorithms in the presence of various quantities on noise. Thus, the use of artificial data serves as a test to see how well the technique detects the known edges. We also construct a similar artificial dataset based on a hyperspectral core image supplied by Geospectral Imaging (Pty) Ltd.

Now we visit the 1-dimensional and the 2-dimensional Fitzhugh-Nagumo equations in order the determine the consistency as well as going through the stability analysis. B. Jacobs has done the contribution in [1] for the following two sections. We visit the analysis and in the next chapter on the development of the proposed 3-dimensional method, a similar analysis will be done.

3.1 Stability of the 1-dimensional scheme

For the one dimensional case, we follow [1]. Suppose we look at the non linear heat equation given by

$$\frac{\partial u}{\partial t} = \mu \frac{\partial^2 u}{\partial^2 x} + \eta u(1-u)(u-\alpha) \tag{3.1}$$

where $0 \leq \mu, \eta, \alpha \leq 1$. This is 1-dimensional as given by one spatial dimension in the model. The partial differential equation has the Crank-Nicholson discretization

$$\frac{u_i^{m+1} - u_i^m}{\Delta t} = \mu \frac{(u_{i+1}^{m+1} - 2u_i^{m+1} + u_{i-1}^{m+1})}{2(\Delta x)^2} + \frac{(u_{i+1}^m - 2u_i^m + u_{i-1}^m)}{2(\Delta x)^2} + \eta f(u_i^m), \tag{3.2}$$

such that $f(u) = u(1-u)(u-\alpha)$. We now multiply through by Δt and bring the $m+1$ terms to the left hand side of equation (3.2) to get

$$-\beta_x u_{i+1}^{m+1} + (1+2\beta_x)u_i^{m+1} - \beta_x u_{i-1}^{m+1} = \beta_x u_{i+1}^m + (1-2\beta_x)u_i^m + u_{i-1}^m + \Delta t \eta f(u_i^m). \tag{3.3}$$

where $\beta_x = \mu \frac{\Delta t}{2(\Delta x)^2}$. Now, using Von Neumann stability analysis [26] on the scheme (3.3), we determine the conditions such that the scheme is stable. Let
$u_i^m = U^m e^{I(\gamma \Delta x i)}$, where $\gamma \in \mathbb{R}$, $I = \sqrt{-1}$. From the scheme, a condition such as $\frac{U^{m+1}}{U^m} < 1$ must be established. In order to atain this condition, firstly, we analyse the quadratic function of the from $f_1(x) = (1-x)(x-\alpha)$ in (3.3), we determine that the extreme points in the function value arise when the derivative function is zero, i.e. $f_1'(x) = 0$. This implies that $-2x + 1 + \alpha = 0$, and thus such point at $x_{max} = \frac{1+\alpha}{2}$. This gives us

$f_1'(x_{max}) = 0$ and substituting into f_1 gives $f_1(x_{max}) = \frac{1}{4}(1-\alpha)^2$, thus

$$f_1(x_{max}) = \begin{cases} 0, & \text{if } \alpha = 1 \\ 1/4, & \text{if } \alpha = 0 \end{cases}$$

and therefore $|f_1(x)| \leq \frac{1}{4}$. It follows, from (3.3) that

$$\begin{aligned} &-\beta_x u_{i+1}^{m+1} + (1+2\beta_x)u_i^{m+1} - \beta_x u_{i+1}^{m+1} \\ &\leq \beta_x u_{i+1}^m + (1-2\beta_x)u_i^m + \beta_x u_{i+1}^m + \Delta t \eta \frac{1}{4} u_i^m \end{aligned} \tag{3.4}$$

Now, we can substitute for $u_i^m = U^m e^{I(\gamma \Delta x i)}$ into (3.4) and rearranging we get that

$$\begin{aligned} &U^{m+1}\left(-\beta_x e^{-I\gamma\Delta x} + (1+2\beta_x) - \beta_x e^{I\gamma\Delta x}\right) \\ &\leq U^m \left(\beta_x e^{-I\rho\Delta x} + (1-2\beta_x) + \beta_x e^{I\rho\Delta x} + \Delta t \eta \frac{1}{4}\right) \end{aligned} \tag{3.5}$$

$$\begin{aligned} \implies &\left|\frac{U^{m+1}}{U^m}\right| \left|-\beta_x e^{-I\gamma\Delta x} + (1+2\beta_x) - \beta_x e^{I\gamma\Delta x}\right| \\ &\leq \left|\beta_x e^{-I\rho\Delta x} + (1-2\beta_x) + \beta_x e^{I\rho\Delta x} + \Delta t \eta \frac{1}{4}\right| \end{aligned} \tag{3.6}$$

$$\implies \left|\frac{U^{m+1}}{U^m}\right| \leq \left|\frac{1 - 4\beta_x sin^2(\frac{\gamma\Delta x}{2}) + \Delta t \eta \frac{1}{4}}{1 + 4\beta_x sin^2\left(\frac{\gamma\Delta x}{2}\right)}\right| \tag{3.7}$$

We require to show that $\left|\frac{U^{m+1}}{U^m}\right| < 1$, we must have

$$\left|\frac{1 - 4\beta_x sin^2(\frac{\gamma\Delta x}{2}) + \Delta t \eta \frac{1}{4}}{1 + 4\beta_x sin^2\left(\frac{\gamma\Delta x}{2}\right)}\right| < 1$$

$$\implies \eta \Delta t < 32\beta_x sin^2\left(\frac{\gamma\Delta x}{2}\right) \tag{3.8}$$

Since $0 \leq sin^2(\theta) \leq 1$, $\forall\theta$ for $0 < \eta\Delta t < 32\beta_x$. Therefore the scheme is stable for $0 < \eta\Delta t < 32\beta_x$.

3.2 Stability of the 2-dimensional scheme

Consider the 2-dimensional nonlinear diffusion equation scheme, a spacial extention of the previous 1-dimensional equation following the development of [1]

$$\frac{\partial u}{\partial t} = \mu\left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y}\right) + \eta f(u) \tag{3.9}$$

Discretized as

$$\frac{u_{i,j}^{m+1} - u_{i,j}^{m}}{\Delta t} = \mu\frac{u_{i+1,j}^{m} - 2u_{i,j}^{m} + u_{i-1,j}^{m}}{(\Delta x)^2} + \mu\frac{u_{i,j+1}^{m} - 2u_{i,j}^{m} + u_{i,j-1}^{m}}{(\Delta y)^2} + \eta f(u_{i,j}^{m}). \tag{3.10}$$

Transforming this into the follwing Crank-Nicholson scheme, becomes

$$\begin{aligned} &- \beta_x u_{i+1}^{m+1} + (1+2\beta_x)u_i^{m+1} - \beta_x u_{i-1}^{m+1} - \beta_y u_{j+1}^{m+1} + (1+2\beta_y)u_j^{m+1} - \beta_y u_{j-1}^{m+1} \\ &= \beta_x u_{i+1}^{m} + (1-2\beta_x)u_i^{m} + \beta_x u_{i-1}^{m} + \beta_y u_{j+1}^{m} + (1-2\beta_y)u_j^{m} + \beta_y u_{j-1}^{m} + \Delta t\eta f(u_j^m). \end{aligned} \tag{3.11}$$

by the Peaceman-Rachford method. Let $u_{i,j}^m = U^m e^{I(\gamma\Delta x i + \rho\Delta y j)}$, where $\gamma, \rho \in \mathbb{R}$ and substitute into (3.11) gives the following expression

$$\begin{aligned} &U^{m+1}(-\beta_x e^{-I\gamma\Delta x} + (1+2\beta_x) - \beta_x e^{I\gamma\Delta x} - \beta_y e^{-I\rho\Delta y} + (1+2\beta_y) + \beta_y e^{I\rho\Delta y}) \\ &= U^m(\beta_x e^{-I\gamma\Delta x} + (1-2\beta_x) + \beta_x e^{I\gamma\Delta x} + \beta_y e^{-I\rho\Delta y} + (1-2\beta_y) + \beta_y e^{I\rho\Delta y}) \\ &+ \Delta t\eta f(u_j^m) \end{aligned} \tag{3.12}$$

$\Longrightarrow$

$$\begin{aligned} &\left|\frac{U^{m+1}}{U^m}\right|\left|1 + 4\beta_x sin^2\left(\frac{\rho\Delta x}{2}\right) + 4\beta_y sin^2\left(\frac{\gamma\Delta y}{2}\right)\right| \\ &\leq \left|1 - 4\beta_x sin^2\left(\frac{\rho\Delta x}{2}\right) - 4\beta_y sin^2\left(\frac{\gamma\Delta y}{2}\right) + \Delta t\eta\frac{1}{4}\right|. \end{aligned} \tag{3.13}$$

$$\left|\frac{U^{m+1}}{U^m}\right| \leq \frac{\left|1 - 4\beta_x sin^2\left(\frac{\rho\Delta x}{2}\right) - 4\beta_y sin^2\left(\frac{\gamma\Delta y}{2}\right) + \Delta t\eta\frac{1}{4}\right|}{\left|1 + 4\beta_x sin^2\left(\frac{\rho\Delta x}{2}\right) + 4\beta_y sin^2\left(\frac{\gamma\Delta y}{2}\right)\right|}. \tag{3.14}$$

By letting $X = \beta_x(sin^2\left(\frac{\rho\Delta x}{2}\right))$and $Y = \beta_y(sin^2\left(\frac{\gamma\Delta y}{2}\right))$, and adding $4\beta_x sin^2\left(\frac{\rho\Delta x}{2}\right) + 4\beta_x sin^2\left(\frac{\rho\Delta x}{2}\right)$ to the numerator we have

$$\left|\frac{U^{m+1}}{U^m}\right| \leq \left|\frac{1 - 4X - 4Y + \Delta t\eta\frac{1}{4}}{1 + 4X + 4Y}\right| \tag{3.15}$$

so that $\left|\frac{U^{m+1}}{U^m}\right| < 1$ if $(1-4X-Y+\frac{\Delta t\eta}{4}) < 1+4X+4Y$. Therefore, since $0 \leq sin^2\left(\frac{\gamma\Delta y}{2}\right) \leq 1$, following the same arguement in the 1-dimensional stability analysis, our stability condition becomes

$$0 < \Delta t\eta < 32(\beta_x + \beta_y). \tag{3.16}$$

The following chapter extends the idea introduced here to 3-dimensional model for the application to hyperspectral data. We go into the purpose of this research to develop the diffusion model proposed.

Chapter 4

3-Dimensional Fitzhugh-Nagumo model

In developing the proposed method, we want to try a similar approach to the study done by Dr B Jacobs and E Momoniat in [1] on text images. In the case of hyperspectral images, the model extends by an extra dimension in the z-axis which means that we need to develop a 3-dimensional Fitzhugh-Nagumo equation. This means that in order to process hyperspectral images, we will treat them as 3-dimensional data matrices. The purpose is to determine how efficient the proposed method is as compared to existing techniques. In contrast to the 2-dimensional model in [1], the diffusion process will occur in all 3 directions simultaneously by the proposed model. Development of this model is the main part of this research and to test how well and efficiently it works.

To begin with the model, first set up the 3-dimensional extension of the model where $u = u(x, y, z, t)$ and the z-component runs along the spectral dimension of the image, where we have the different image bands. This potentially allows us to find edges in the entire hyperspectral data cube rather than in just a 2-dimensional slice by taking advantage of the spectral differences in the image. As part of this research, techniques in computational differential equations for solving partial differential equations, specifically discretization of the PDE and stability check will be applied here on the proposed model as done in the previous chapter.

Consider the nonlinear three dimensional diffusion equation,

$$\frac{\partial u}{\partial t} = \mu \left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y} + \frac{\partial^2 u}{\partial^2 z} \right) + \eta u(1-u)(u-\alpha), \tag{4.1}$$

$$u(x, y, z, 0) = u_0, \tag{4.2}$$

$$u_x(0, y, z, t) = 0, u_x(x_N, y, z, t) = 0, \tag{4.3}$$

$$u_y(x, 0, z, t) = 0, u_y(x, y_N, z, t) = 0, \tag{4.4}$$

$$u_z(x, y, 0, t) = 0, u_z(x, y, z_N, t) = 0, \tag{4.5}$$

where the initial condition u_0 is the original unprocessed image. For the purpose of this research, we use the reflective boundary condition. This means that at the boundaries, for all values $t > 0$, pixel value u in the diffusion equation represents the "energy" of the pixels and does not escape from the image domain. Thus the diffused pixel values are reflected back into the images at these boundaries. The model ensures that we are only working within the space that is the image and ensures that the diffusion of these values remains within the image "space" at each of these image boundaries. Hence the derivatives of u at the boundaries $\{(0, y, z), (x, 0, z), (x, y, 0)\}, \{(x_N, y, z), (x, y_N, z), (x, y, z_N)\}$ in all the directions are zero.

In order to further understand the nature of this equation, we explore briefly the nonlinear term of the diffusion model. Each of the plots below demonstrates the shape of the nonlinear term for different values of α.

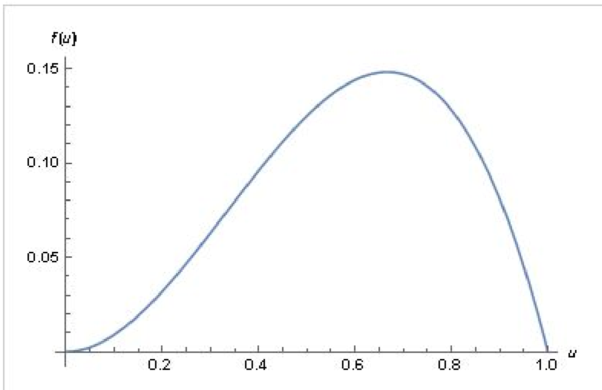


Figure 4.1: Shape of nonlinear term with $\alpha = 0$.

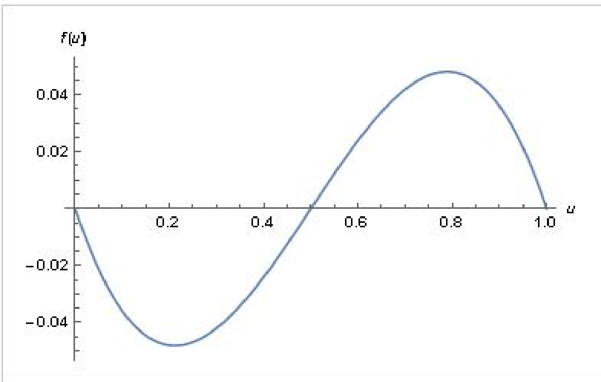


Figure 4.2: Shape of nonlinear term with $\alpha = 0.5$.

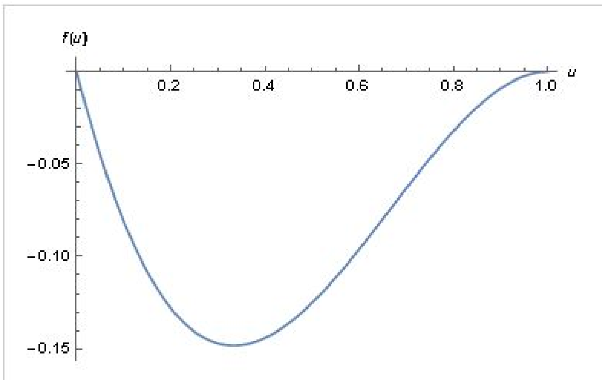


Figure 4.3: Shape of nonlinear term with $\alpha = 1$.

The partial differential equation is to be solved numerically by discretization, which is ideal for visualization of the resulting image as each discrete element of the image represents a pixel value in a band of the image. The solution of the PDE by discretization is analyzed and its consistency to the exact solution of the PDE is to be determined. The domain of the image is discretized by a grid over the image and the step size in the x-directions is given by $\Delta x = \frac{1}{N_x}$, where N_x is the number of pixels in the x-directions. Similarly for the y-, z-direction and also for the time step Δt, for which the conditions will be determined later. A pixel in the image can now be considered in the grid as $u(x_i, y_j, z_k, t_m)$, where $i = 0, 1, 2, ..., N_x, j = 0, 1, 2, ..., N_y, k = 0, 1, 2, ..., N_z$ and $m = 0, 1, 2, ..., N_t$. Now, a Taylor expansion approximation of the first and the second derivative can be expressed as

$$u_x = \lim_{\Delta x \to 0} \frac{u(x_{i+1}, y_j, z_k, t_m) - u(x_i, y_j, z_k, t_m)}{\Delta x}, \tag{4.6}$$

$$u_{xx} = \lim_{\Delta x \to 0} \frac{u(x_{i+1}, y_j, z_k, t_m) - 2u(x_i, y_j, z_k, t_m) + u(x_{i-1}, y_j, z_k, t_m)}{(\Delta x)^2}. \tag{4.7}$$

Throughout the paper, these approximations, for instance u_x, will be written as,

$$u_x \approx \frac{u^m_{i+1,j,k} - u^m_{i,j,k}}{\Delta x}. \tag{4.8}$$

A discretization of (4.1) is,

$$\begin{aligned} \frac{u^{m+1}_{i,j,k} - u^m_{i,j,k}}{\Delta t} &= \mu \frac{u^m_{i+1,j,k} - 2u^m_{i,j,k} + u^m_{i-1,j,k}}{(\Delta x)^2} + \mu \frac{u^m_{i,j+1,k} - 2u^m_{i,j,k} + u^m_{i,j-1,k}}{(\Delta y)^2} \\ &+ \mu \frac{u^m_{i,j,k+1} - 2u^m_{i,j,k} + u^m_{i,j,k-1}}{(\Delta z)^2} + \eta f(u^m_{i,j,k}), \end{aligned} \tag{4.9}$$

where $f(u^m_{i,j,k}) = u^m_{i,j,k}(1 - u^m_{i,j,k})(u^m_{i,j,k} - \alpha)$ and α is the threshold value which will enable highlighting certain pixel values by shifting them from the background. Now let $\beta_x = \frac{\mu \Delta t}{3(\Delta x)^2}$, and similarly for β_y and β_z. Δt, Δx, Δy and Δz are the respective step lengths in time and spacially. By rearranging (4.9) we obtain,

$$\begin{aligned} u^{m+1}_{i,j,k} = u^m_{i,j,k} &+ \beta_x(u^m_{i+1,j,k} - 2u^m_{i,j,k} + u^m_{i-1,j,k}) + \beta_y(u^m_{i,j+1,k} - 2u^m_{i,j,k} + u^m_{i,j-1,k}) \\ &+ \beta_z(u^m_{i,j,k+1} - 2u^m_{i,j,k} + u^m_{i,j,k-1}) + \Delta t \eta f(u^m_{i,j,k}). \end{aligned} \tag{4.10}$$

Since this is a discretization of the equation in 3-D, a suitable method of solving (4.10) is by the use of an alternating - direction implicit (ADI) method. A 2-D approach was implemented in paper [1]. Using a similar Crank-Nicholson with Peaceman-Rachford ADI method [26] to develop the computational scheme in 3-D, we end up with the three stage scheme

$$(1 - \beta_x \delta_x^2) u^{m+\frac{1}{3}}_{i,j,k} = (1 + \beta_y \delta_y^2 + \beta_z \delta_z^2) u^m_{i,j,k} + \Delta t \eta f(u^m_{i,j,k}), \tag{4.11}$$

$$(1 - \beta_y \delta_y^2) u^{m+\frac{2}{3}}_{i,j,k} = (1 + \beta_x \delta_x^2 + \beta_z \delta_z^2) u^{m+\frac{1}{3}}_{i,j,k} + \Delta t \eta f(u^{m+\frac{1}{3}}_{i,j,k}), \tag{4.12}$$

$$(1 - \beta_z \delta_z^2) u^{m+1}_{i,j,k} = (1 + \beta_x \delta_x^2 + \beta_y \delta_y^2) u^{m+\frac{2}{3}}_{i,j,k} + \Delta t \eta f(u^{m+\frac{2}{3}}_{i,j,k}), \tag{4.13}$$

where $\delta_l^2 u_l^m = u^m_{l+1} - 2u^m_l + u^m_{l-1}$ for each of $l = x, y, z$ in each of the corresponding non-time directions.

4.1 Stability of the 3-dimensional scheme

Now, to determine the stability of the 3-dimensional scheme we employ the same method when analysing the stability of the 1-dimensional and the 2-dimensional schemes. Thus, Let $u_{i,j}^m = U^m e^{I(\gamma\Delta xi+\rho\Delta yj+\omega\Delta zk)}$, where $\gamma, \rho, \omega \in \mathbb{R}$ and substitute for the Crank-Nicholson scheme. We omit indices in which we are not differentiating, i.e. for $u_{i+1,j,k}^{m+1}$ we write u_{i+1}^{m+1} and for $u_{i,j,k+1}^{m+1}$ we write u_{k+1}^{m+1} and so on.

$$\begin{aligned}
&-\beta_x u_{i+1}^{m+1} + (1+2\beta_x)u_i^{m+1} - \beta_x u_{i-1}^{m+1} - \beta_y u_{j+1}^{m+1} + (1+2\beta_y)u_j^{m+1} - \beta_y u_{j-1}^{m+1}\\
&-\beta_z u_{k+1}^{m+1} + (1+2\beta_z)u_k^{m+1} - \beta_z u_{k-1}^{m+1}\\
&= \beta_x u_{i+1}^{m} + (1-2\beta_x)u_i^{m} + \beta_x u_{i-1}^{m} + \beta_y u_{j+1}^{m} + (1-2\beta_y)u_j^{m} + \beta_y u_{j-1}^{m}\\
&\beta_z u_{k+1}^{m+1} + (1-2\beta_z)u_k^{m+1} + \beta_z u_{k-1}^{m+1} + \Delta t\eta f(u_j^m).
\end{aligned} \tag{4.14}$$

$$\left|\frac{U^{m+1}}{U^m}\right| \le \left|\frac{1-4\beta_x sin^2\left(\frac{\rho\Delta x}{2}\right) - 4\beta_y sin^2\left(\frac{\gamma\Delta y}{2}\right) - 4\beta_z sin^2\left(\frac{\omega\Delta z}{2}\right) + \Delta t\eta\frac{1}{4}}{1+4\beta_x sin^2\left(\frac{\rho\Delta x}{2}\right) + 4\beta_y sin^2\left(\frac{\gamma\Delta y}{2}\right) + 4\beta_z sin^2\left(\frac{\omega\Delta z}{2}\right)}\right|. \tag{4.15}$$

let $X = \beta_x sin^2\left(\frac{\rho\Delta x}{2}\right), Y = \beta_y sin^2\left(\frac{\gamma\Delta y}{2}\right)$ and $Z = \beta_z sin^2\left(\frac{\omega\Delta z}{2}\right)$. Equation (4.15) results in

$$\left|\frac{U^{m+1}}{U^m}\right| \le \left|\frac{1-4X-4Y-4Z+\Delta t\eta\frac{1}{4}}{1+4X+4Y+4Z}\right|, \tag{4.16}$$

since $0 \le \beta_x sin(\frac{\rho\Delta x}{2}), \beta_y sin(\frac{\gamma\Delta y}{2}), \beta_z sin(\frac{\omega\Delta z}{2}) \le \beta_x, \beta_y, \beta_z$, respectively. The desired condition must satisfy $|\frac{U^{m+1}}{U^m}| < 1$ for stability. Therefore this is true if

$$\Delta t\eta\frac{1}{4} < 8X + 8Y + 8Z \tag{4.17}$$

$$\implies \Delta t\eta < 32(\beta_x + \beta_y + \beta_z) \tag{4.18}$$

We notice the following expression the n-th order Fitzhugh-Nagumo equations studied here.

$$\Delta t\eta < 32\sum_{i=1}^{n}\beta_i \tag{4.19}$$

where $n = 1, 2, 3,$ is the number of dimensions of a particular problem thus, $n = 1$ is the first dimension, $n = 2$ is the second dimension and $n = 3$ is the third dimension. Of course, these conditions are determined by the physical nature of the problem, in this case the image itself. This makes it possible as we assumed in the calculations for $\left|\frac{U^{m+1}}{U^m}\right| < 1$.

4.2 On the Neumann Boundary Conditions

On the note about the boundary condition of the diffusion problem (4.1), we have the following:
we employ the central difference discretization for the spatial dimensional, such as (4.8) and forwards difference is time. Equation (4.3) gives

$$u_x = \frac{\partial u}{\partial x}, \tag{4.20}$$

$$\frac{u^m_{i+1,j,k} + u^m_{i-1,j,k}}{\Delta x} = 0, \tag{4.21}$$

where, $i = 0$ by the chosen boundary conditions.

$$u^m_{1,j,k} = u^m_{-1,j,k}. \tag{4.22}$$

Since $u^m_{-1,j,k}$ is a fictitious point, it is then replaced by $u^m_{1,j,k}$, accounted for by the nature of the boundary conditions when substituting the point in the scheme matrix $\mathbf{A_x}$ in (4.49). Thus, (4.10) implies

$$\begin{aligned} u^{m+1}_{0,j,k} = u^m_{0,j,k} + \beta_x u^m_{1,j,k} - 2\beta_x u^m_{0,j,k} + \beta_x u^m_{-1,j,k} + \beta_y u^m_{0,j+1,k} - 2\beta_y u^m_{0,j,k} + \beta_y u^m_{0,j-1,k} \\ + \beta_z u^m_{0,j,k+1} - \beta_z 2u^m_{0,j,k} + \beta_z u^m_{0,j,k-1} + \Delta t \eta f(u^m_{0,j,k}). \end{aligned} \tag{4.23}$$

$$\begin{aligned} u^{m+1}_{0,j,k} = (1 - 2\beta_x) u^m_{0,j,k} + 2\beta_x u^m_{1,j,k} + \beta_y (u^m_{0,j+1,k} - 2u^m_{0,j,k} + u^m_{0,j-1,k}) \\ + \beta_z (u^m_{0,j,k+1} - 2u^m_{0,j,k} + u^m_{0,j,k-1}) + \Delta t \eta f(u^m_{0,j,k}). \end{aligned} \tag{4.24}$$

Now, when $i = N$ the second boundary in the x-direction accounts for the fictitious point by the following result:

$$u^m_{N+1,j,k} = u^m_{N-1,j,k}. \tag{4.25}$$

Also, we have the expression of the second boundary condition in the x-direction:

$$\begin{aligned} u^{m+1}_{N,j,k} = 2\beta_x u^m_{N-1,j,k} + (1 - 2\beta_x) u^m_{N,j,k} + \beta_y (u^m_{N,j+1,k} - 2u^m_{N,j,k} + u^m_{N,j-1,k}) \\ + \beta_z (u^m_{N,j,k+1} - 2u^m_{N,j,k} + u^m_{N,j,k-1}) + \Delta t \eta f(u^m_{N,j,k}). \end{aligned} \tag{4.26}$$

In a similar fashion, it can be concluded that the boundary conditions in the y- and the z-directions derived from equations (4.4) and (4.5) can be expressed as follows

$$\begin{aligned} u_{i,0,k}^{m+1} &= (1-2\beta_y)u_{i,0,k}^m + 2\beta_y u_{i,N_y,k}^m + \beta_x(u_{i+1,0,k}^m - 2u_{i,0,k}^m + u_{i-1,0,k}^m) \\ &\quad + \beta_z(u_{i,0,k+1}^m - 2u_{i,0,k}^m + u_{i,0,k-1}^m) + \Delta t\eta f(u_{i,0,k+1}^m). \end{aligned} \tag{4.27}$$

$$\begin{aligned} u_{i,N_y,k}^{m+1} &= 2\beta_y u_{i,N_y-1,k}^m + (1-2\beta_y)u_{i,N_y,k}^m + \beta_x(u_{i+1,N_y,k}^m - 2u_{i,N_y,k}^m + u_{i-1,N_y,k}^m) \\ &\quad + \beta_z(u_{i,N_y,k+1}^m - 2u_{i,N_y,k}^m + u_{i,N_y,k-1}^m) + \Delta t\eta f(u_{i,N_y,k}^m), \end{aligned} \tag{4.28}$$

$$\begin{aligned} u_{i,j,0}^{m+1} &= (1-2\beta_z)u_{i,j,0}^m + 2\beta_z u_{i,j,1}^m + \beta_x(u_{i+1,j,0}^m - 2u_{i,j,0}^m + u_{i-1,j,0}^m) \\ &\quad + \beta_y(u_{i,j+1,0}^m - 2u_{i,j,0}^m + u_{i,j-1,0}^m) + \Delta t\eta f(u_{i,j,0}^m). \end{aligned} \tag{4.29}$$

$$\begin{aligned} u_{i,j,N_z}^{m+1} &= 2\beta_z u_{i,,j,N_z-1}^m + (1-2\beta_z)u_{i,j,N_z}^m + \beta_x(u_{i+1,k,N_z}^m - 2u_{i,j,N_z}^m + u_{i-1,j,N_z}^m) \\ &\quad + \beta_y(u_{i,j+1,N_z}^m - 2u_{i,j,N_z}^m + u_{i,j-1,N_z}^m) + \Delta t\eta f(u_{i,j,N_z}^m). \end{aligned} \tag{4.30}$$

With these equations, the boundaries can be accounted for within the matrices explained in the next section that results from the schemes in (4.11) - (4.13), which are used in the MatLab code for the proposed model.

4.3 Consistency of the 3-dimensional Fitzhugh-Nagumo scheme

We recall the discretized scheme

$$\begin{aligned} \frac{u_{i,j,k}^{m+1} - u_{i,j,k}^m}{\Delta t} &= \frac{u_{i+1,j,k}^{m+1} - 2u_{i,j,k}^{m+1} + u_{i-1,j,k}^{m+1}}{2(\Delta x)^2} + \frac{u_{i,j+1,k}^{m+1} - 2u_{i,j,k}^{m+1} + u_{i,j-1,k}^{m+1}}{2(\Delta y)^2} \\ &\quad + \frac{u_{i,j,k+1}^{m+1} - 2u_{i,j,k}^{m+1} + u_{i,j,k-1}^{m+1}}{2(\Delta z)^2} + \frac{u_{i+1,j,k}^m - 2u_{i,j,k}^m + u_{i-1,j,k}^m}{2(\Delta x)^2} \\ &\quad + \frac{u_{i,j+1,k}^m - 2u_{i,j,k}^m + u_{i,j-1,k}^m}{2(\Delta y)^2} + \frac{u_{i,j,k+1}^m - 2u_{i,j,k}^m + u_{i,j,k-1}^m}{2(\Delta z)^2} + f(u_{i,j,k}^m), \end{aligned} \tag{4.31}$$

and that the approximation is given by u_x

$$u_x \approx \frac{u_{i,j,k}^{m+1} - u_{i,j,k}^m}{\Delta x}. \tag{4.32}$$

We want to show that the approximation solution of the scheme converges to the solution of the Fitzhu-Naghumo partial differential equation. We take each of the terms in the approximation equation (4.31) individually. Expanding each of the approximations by Taylor Series:

$$u_{i+1}^{m+1} = u^{m+1} + \Delta x u_x^{m+1} + \frac{(\Delta x)^2}{2} u_{xx}^{m+1} + \frac{(\Delta x)^3}{3!} u_{xxx}^{m+1} + O((\Delta x)^4) \tag{4.33}$$

$$u_{i-1}^{m+1} = u^{m+1} - \Delta x u_x^{m+1} + \frac{(\Delta x)^2}{2} u_{xx}^{m+1} - \frac{(\Delta x)^3}{3!} u_{xxx}^{m+1} + O((\Delta x)^4) \tag{4.34}$$

Now expanding about t:

$$\begin{aligned} u_{i+1}^{m+1} =& u + \Delta x u_x + \frac{(\Delta x)^2}{2} u_{xx} + O((\Delta x)^3) \\ &+ \Delta t \left(u_t + \Delta x u_{xt} + \frac{(\Delta x)^2}{2} u_{xxt} + O((\Delta x)^3) \right) \\ &+ \frac{(\Delta t)^2}{2} \left(u_{tt} + \Delta x u_{ttx} + \frac{(\Delta x)^2}{2} u_{xxtt} + O((\Delta x)^3) \right) \\ &+ O((\Delta t)^3) \end{aligned} \tag{4.35}$$

Similarly:

$$\begin{aligned} u_{i-1}^{m+1} =& u - \Delta x u_x + \frac{(\Delta x)^2}{2} u_{xx} - O((\Delta x)^3) \\ &+ \Delta t \left(u_t - \Delta x u_{xt} + \frac{(\Delta x)^2}{2} u_{xxt} - O((\Delta x)^3) \right) \\ &+ \frac{(\Delta t)^2}{2} \left(u_{tt} - \Delta x u_{ttx} + \frac{(\Delta x)^2}{2} u_{xxtt} + O((\Delta x)^3) \right) \\ &+ O((\Delta t)^3) \end{aligned} \tag{4.36}$$

Now

$$u^{m+1} = u + \Delta t u_t + \frac{(\Delta t)^2}{2} u_{tt} + \frac{(\Delta t)^3}{3!} u_{ttt} + O((\Delta x)^3) \tag{4.37}$$

Adding (4.35), (4.36) and (4.37) as in the approximation scheme, we get:

$$
\begin{aligned}
u_{i-1}^{m+1} - 2u^{m+1} + u_{i+1}^{m+1} &= u - \Delta x u_x + \frac{(\Delta x)^2}{2} u_{xx} + O((\Delta x)^3) + \Delta t u_t - \Delta t \Delta x u_{xt} \\
&+ \frac{\Delta t (\Delta x)^2}{2} u_{txx} + O((\Delta x)^3 \Delta t) + \frac{(\Delta t)^2}{2} u_{tt} - \frac{(\Delta t)^2 \Delta x}{2} u_{ttx} \\
&+ \frac{(\Delta x)^2 (\Delta t)^2}{4} u_{xxtt} + O((\Delta t)^2 (\Delta x)^3) - 2u - 2\Delta t u_t - 2\frac{(\Delta t)^2}{2} u_{tt} \\
&- \frac{(\Delta t)^3}{3} u_{ttt} + O((\Delta t)^4) + u + \Delta x u_x + \frac{(\Delta x)^2}{2} u_{xx} + O((\Delta x)^3) + \Delta t u_t \\
&+ \Delta x \Delta t u_{xt} + \frac{\Delta t (\Delta x)^2}{2} u_{txx} + O((\Delta x)^3 \Delta t) + \frac{(\Delta t)^2}{2} u_{tt} \\
&+ \frac{(\Delta t)^2 \Delta x}{2} u_{ttx} + \frac{(\Delta x)^2 (\Delta t)^2}{4} u_{xxtt} + O((\Delta t)^2 ((\Delta x)^3)
\end{aligned}
\tag{4.38}
$$

$\Longrightarrow$

$$
\begin{aligned}
u_{i-1}^{m+1} - 2u^{m+1} + u_{i+1}^{m+1} &= (\Delta x)^2 u_{xx} + \Delta t (\Delta x)^2 u_{txx} \\
&+ \frac{(\Delta x)^2 (\Delta t)^2}{2} u_{xxtt} + O(\Delta t (\Delta x)^3) + O((\Delta t)^2 (\Delta x)^3) + O((\Delta x)^3)
\end{aligned}
\tag{4.39}
$$

$\Longrightarrow$

$$
\begin{aligned}
\frac{u_{i-1}^{m+1} - 2u^{m+1} + u_{i+1}^{m+1}}{2(\Delta x)^2} &= \frac{1}{2} u_{xx} + \frac{\Delta t}{2} u_{txx} \\
&+ \frac{(\Delta t)^2}{4} u_{xxtt} + O(\Delta t \Delta x) + O((\Delta t)^2 \Delta x) + O(\Delta x)
\end{aligned}
\tag{4.40}
$$

Similary, we obtain the same expression in the y and z directions, respectively:

$$
\begin{aligned}
\frac{u_{j-1}^{m+1} - 2u_j^{m+1} + u_{j+1}^{m+1}}{2(\Delta y)^2} &= \frac{1}{2} u_{yy} + \frac{\Delta t}{2} u_{tyy} \\
&+ \frac{(\Delta t)^2}{4} u_{yytt} + O(\Delta t \Delta y) + O((\Delta t)^2 \Delta y) + O(\Delta y)
\end{aligned}
\tag{4.41}
$$

$$
\begin{aligned}
\frac{u_{k-1}^{m+1} - 2u_k^{m+1} + u_{k+1}^{m+1}}{2(\Delta z)^2} &= \frac{1}{2} u_{zz} + \frac{\Delta t}{2} u_{tzz} \\
&+ \frac{(\Delta t)^2}{4} u_{zztt} + O(\Delta t \Delta z) + O((\Delta t)^2 \Delta z) + O(\Delta z)
\end{aligned}
\tag{4.42}
$$

Now expanding the approximation at time step m, as done above for time step $m+1$ we get the following:

$$\begin{aligned}\frac{u_{i-1}^{m}-2u_{i}^{m}+u_{i+1}^{m}}{2(\Delta x)^2} &= \frac{1}{2(\Delta x)^2}\left(u+\Delta x u_x+\frac{(\Delta x)^2}{2}u_{xx}+O((\Delta x)^3)\right)\\ &+\frac{1}{2(\Delta x)^2}\left(-2u+u-\Delta x u_x+\frac{(\Delta x)^2}{2}u_{xx}+O((\Delta x)^3)\right)\end{aligned} \tag{4.43}$$

$\Longrightarrow$

$$\frac{u_{i-1}^{m}-2u_{i}^{m}+u_{i+1}^{m}}{2(\Delta x)^2}=\frac{1}{2}u_{xx}+O((\Delta x)^2) \tag{4.44}$$

Adding (4.40) and (4.44) together:

$$\begin{aligned}&\frac{u_{k-1}^{m+1}-2u_{k}^{m+1}+u_{k+1}^{m+1}}{2(\Delta x)^2}+\frac{u_{i-1}^{m}-2u_{i}^{m}+u_{i+1}^{m}}{2(\Delta x)^2}=u_{xx}+\frac{\Delta t}{2}u_{txx}\\ &+\frac{(\Delta t)^2}{4}u_{xxtt}+O(\Delta t\Delta x)+O((\Delta t)^2\Delta x)+O(\Delta x)+O((\Delta x)^2)\end{aligned} \tag{4.45}$$

Now by taking the limits as Δt and Δx approaching zero, the left hand side of (4.45) become

$$\lim_{\Delta x\to 0}\left(\frac{u_{i-1}^{m+1}-2u_{i}^{m+1}+u_{i+1}^{m+1}}{2(\Delta x)^2}+\frac{u_{i-1}^{m}-2u_{i}^{m}+u_{i+1}^{m}}{2(\Delta x)^2}\right)=u_{xx} \tag{4.46}$$

similarly for y and z directions derived on (4.41) and (4.45)

$$\lim_{\Delta y\to 0}\left(\frac{u_{j-1}^{m+1}-2u_{j}^{m+1}+u_{j+1}^{m+1}}{2(\Delta y)^2}+\frac{u_{j-1}^{m}-2u_{j}^{m}+u_{j+1}^{m}}{2(\Delta y)^2}\right)=u_{yy} \tag{4.47}$$

$$\lim_{\Delta z\to 0}\left(\frac{u_{k-1}^{m+1}-2u^{m+1}+u_{k+1}^{m+1}}{2(\Delta z)^2}+\frac{u_{k-1}^{m}-2u_{k}^{m}+u_{k+1}^{m}}{2(\Delta z)^2}\right)=u_{zz} \tag{4.48}$$

Thus the scheme is convergent to the original partial differential equation as all terms in the scheme in the z, y and x-direction converge to the original second order derivatives of the PDE.

4.4 Execution of the 3-dimensional scheme

Since the implementation of the 1- and the 2-dimensional cases are discussed by Jacobs and Momoniat in [1], we look into the 3-dimensional case. We recall (4.11), (4.12) and (4.13) for computational purposes, and consider the following system by rearranging,

$$\mathbf{A_x u^{m+\frac{1}{3}}} = \mathbf{B_y u^m} + \mathbf{C_z u^m} + \mathbf{f(u^m)}, \tag{4.49}$$

$$\mathbf{A_y u^{m+\frac{2}{3}}} = \mathbf{B_x u^m} + \mathbf{C_z u^m} + \mathbf{f(u^m)}, \tag{4.50}$$

$$\mathbf{A_z u^{m+\frac{2}{3}}} = \mathbf{B_x u^m} + \mathbf{C_y u^m} + \mathbf{f(u^m)}, \tag{4.51}$$

where $\mathbf{f(u^m)}$ is the source term,

$$\mathbf{A_x} = \begin{pmatrix} (1+2\beta_x) & -2\beta_x & 0 & 0 & \cdots & 0 \\ -\beta_x & (1+2\beta_x) & -\beta_x & 0 & \cdots & 0 \\ 0 & -\beta_x & (1+2\beta_x) & -\beta_x & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & -\beta_x & (1+2\beta_x) & -\beta_x \\ 0 & \cdots & & 0 & -2\beta_x & (1+2\beta_x) \end{pmatrix}, \tag{4.52}$$

$$\mathbf{B_y} = \begin{pmatrix} (1-2\beta_y) & \beta_y & 0 & 0 & \cdots & 0 \\ 2\beta_y & (1-2\beta_y) & \beta_y & 0 & \cdots & 0 \\ 0 & \beta_y & (1-2\beta_y) & \beta_y & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & \beta_y & (1-2\beta_y) & 2\beta_y \\ 0 & \cdots & & 0 & \beta_y & (1-2\beta_y) \end{pmatrix}, \tag{4.53}$$

$$
\mathbf{C_z} = \begin{pmatrix}
-2\beta_z & 2\beta_z & 0 & 0 & \cdots & 0 \\
\beta_z & -2\beta_z & \beta_z & 0 & \cdots & 0 \\
0 & \beta_z & -2\beta_z & \beta_z & \ddots & \vdots \\
\vdots & 0 & \ddots & \ddots & \ddots & 0 \\
0 & & \ddots & \beta_z & -2\beta_z & \beta_z \\
0 & \cdots & & 0 & 2\beta_z & -2\beta_z
\end{pmatrix}, \tag{4.54}
$$

$$
\mathbf{A_y} = \begin{pmatrix}
(1+2\beta_y) & -\beta_y & 0 & 0 & \cdots & 0 \\
-2\beta_y & (1+2\beta_y) & -\beta_y & 0 & \cdots & 0 \\
0 & -\beta_y & (1+2\beta_y) & -\beta_y & \ddots & \vdots \\
\vdots & 0 & \ddots & \ddots & \ddots & 0 \\
0 & & \ddots & -\beta_y & (1+2\beta_y) & -2\beta_y \\
0 & \cdots & & 0 & -\beta_y & (1+2\beta_y)
\end{pmatrix}, \tag{4.55}
$$

$$
\mathbf{B_x} = \begin{pmatrix}
(1-2\beta_x) & 2\beta_x & 0 & 0 & \cdots & 0 \\
\beta_x & (1-2\beta_x) & \beta_x & 0 & \cdots & 0 \\
0 & \beta_x & (1-2\beta_x) & \beta_x & \ddots & \vdots \\
\vdots & 0 & \ddots & \ddots & \ddots & 0 \\
0 & & \ddots & \beta_x & (1-2\beta_x) & \beta_x \\
0 & \cdots & & 0 & 2\beta_x & (1-2\beta_x)
\end{pmatrix}, \tag{4.56}
$$

$$
\mathbf{A_z} = \begin{pmatrix}
(1+2\beta_z) & -2\beta_z & 0 & 0 & \cdots & 0 \\
-\beta_z & (1+2\beta_z) & -\beta_z & 0 & \cdots & 0 \\
0 & -\beta_x & (1+2\beta_z) & -\beta_z & \ddots & \vdots \\
\vdots & 0 & \ddots & \ddots & \ddots & 0 \\
0 & & \ddots & -\beta_z & (1+2\beta_z) & -\beta_z \\
0 & \cdots & & 0 & -2\beta_z & (1+2\beta_z)
\end{pmatrix}, \tag{4.57}
$$

$$
\mathbf{C_y} = \begin{pmatrix}
-2\beta_y & \beta_z & 0 & 0 & \cdots & 0 \\
2\beta_z & -2\beta_y & \beta_y & 0 & \cdots & 0 \\
0 & \beta_y & -2\beta_y & \beta_y & \ddots & \vdots \\
\vdots & 0 & \ddots & \ddots & \ddots & 0 \\
0 & & \ddots & \beta_y & -2\beta_y & 2\beta_y \\
0 & \cdots & & 0 & \beta_y & -2\beta_y
\end{pmatrix}. \tag{4.58}
$$

Matrices (4.52) - (4.58) are the operator matrices in the MatLab code applied onto the image matrix $\mathbf{u^m}$ in order for the diffusion process to be achieved. The respective first row and last row, first column and last column are due to the fact that we have Neumann boundary conditions in the respective matrices. The i-th row of these matrices is equal to the i-th component in the x-direction of the image, similarly for y- and z-direction. Therefore, they are square matrices. At each iteration m, equations (4.49), (4.50) and (4.51) update each of $(\mathbf{u^{m+\frac{1}{3}}})$, $(\mathbf{u^{m+\frac{2}{3}}})$ and $(\mathbf{u^{m+1}})$ according to the Peaceman-Rachford scheme. Matrices $\mathbf{A_x}$ and $\mathbf{A_y}$ differ in that the x-direction is chosen in the vertical direction and the y-direction is chosen in the horizontal direction.

Taking note that in this case, the image is a 3-dimensional matrix in MatLab, in order to make matrix computations easier in the z-direction, the image was treated as a series 2-dimensional matrices when processing. This also made it possible to produce the matrices $\mathbf{C_z}$ and $\mathbf{A_z}$, which facilitates the diffusion process in the z-direction. The equations (4.49) - (4.51) is a step by step process which shows that processing as a series of two dimensions would be the same processing once in the entire image. This is because firstly we process the in the x-direction described by (4.49), once done we move to the y-direction as processing described by (4.49) and so on. Now by permutation of the three dimensional image matrix, we bring the third direction (z-direction) to the x-direction and multiply with the respective $\mathbf{C_z}$ and $\mathbf{A_z}$. We convert back after the computations at each iteration. It makes it possible for the pixel "energy" at each band to diffuse to neighbouring band within the same pixel. Therefore, the entire diffusion process is applied in 3-dimensions.

The FN technique iteratively solves for the values u_{i+1} based on previous u_i making use of the neighbouring pixel values in the x, y and z directions as time evolves in discretized time steps of length m, where pixel value u can take up the value 0 or 1 as time evolves due to the source term within the proposed model equation (4.1).

Chapter 5

Results

In this section, we will be looking at the application of the various methods mentioned to be tested. We begin by giving a short description of how we produced the artificial data and then give an analysis of results from each of the methods that were tested. We then applied this proposed method on an RGB image of a coin, which only has three bands. We then look at hyperspectral images which in contrast to the RGB image, was hundreds of bands as described in the introduction.

The artificial test data sets used for this research have been produced using MatLab. An artificial Cuprite dataset is based on the real Cuprite Nevada hyperspectral image acquired by AVIRIS [27]. This dataset was introduced in [24]. The image was created using $K = 29$ endmembers selected from USGS spectral library. From the real Cuprite AVIRIS image, an abundance matrix of the endmembers was derived to create an artificial hyperspectral image used in the results section [6]. The artificial image contains 350 by 350 pixels with 181 bands. Using the Principal Component Analysis has helped to be able to visualize the different compositions of each endmember. It also helped to further analyze the processed results in this paper and compare them.

Using a similar approach as described above, we created the artificial core imager data set. The process involves estimating the noise in the imager data with the Roger's second method for noise estimation [28]. After estimating noise within the image, the imager data was estimated to have a composition of nine endmembers. Now, using the

NFINDR method together with the estimated number of endmembers, we determine the actual nine endmembers that explain the makeup of the data. The abundance matrix obtained by unmixing against the original endmembers is then used to create an artificial core image by remixing against the chosen nine endmembers. The techniques to be tested are supposed to give expected results for the artificial test data because we know what it contains and where the edges are. Results are expected to give a good approximation if not the exact location of edges for the different regions in the real data also.

5.1 Exploring the Fitzhugh-Nagumo method on a normal RGB image

We start by doing a test on a three dimensional image data set which is a normal RGB image, having only three bands. Figure (5.1a) is an RGB image of this form, figure (5.1b) has added *Speckle* noise from MatLab. When the diffusion model is applied to the coin image, the resulting output shown in figure (5.2) and in addition to the diffusion, highlighting and low-lighting results, as seen. It clearly shows that the noise was diffused and completely removed while the structure and features of the coin are highlighted. Since the background of the image is brighter than the coin, it has become completely white, hence the brighter values, an effect of the source term. The coin image is a much smaller data set compared to the hyperspectral images. Instead of having hundreds of bands, it only has 3 bands that are the Red, Blue and Green colours.

(a) (b)

Figure 5.1: (a) figure showing an image of a coin and (b) with *Speckle* noise density set to 0.004 in MatLab.

Figure 5.2: result of the proposed method for $\mu = 0.4$, $\eta = 1$ after 3 iterations on the coin image.

(a) (b)

Figure 5.3: Figure (a) shows edge detection on the results of the FN method of the image with noise, (b) result of the canny edge detection.

We see here that the proposed method removes the noise presented in this coin image

successfully while enhancing the main features within it. The words "ONE PENNY" as well as the grid like structure remains as it was and are more enhanced. With the canny edge detection from MatLab, the writing can be clearly seen but most of the noise around the coin was also detected as edges. This is taken care of in the proposed method by the source term in the model.
Now we analyse the proposed method against some of the other techniques introduced in the literature review chapter.

Firstly, the initial method to be tested is the proposed Fitzhugh-Nagumo model. It is applied to the artificial Cuprite hyperspectral image to enhance the regions and determine how they are separated and then compared to the unprocessed original data. In order to visualize this data, we looked at a slice at the particular band of interest. We also made use of the Principal component analysis (PCA) (explained in [29]), where we use the entire data to give the best image representation in 3 bands for both the processed and the unprocessed data, thus compressing the information from all the bands into just a "normal" picture.

We used MatLab to develop and run the Fitzhugh-Nagumo model on the hyperspectral dataset. To begin the process, the image is imported into MatLab and converted into a 3-dimensional matrix, normalized between values of 0 and 1. This is our initial condition, u_0 to solve the PDE (4.1). The numerical scheme derived based on the Fitzhugh-Nagumo was applied with a few different numbers of iterations for the diffusion process on the input matrix datasets. This is to test and show the effects of the different parameters used within the model. This will be shown later.

5.2 Fitzhugh-Nagumo method applied to artificial hyperspectral core image

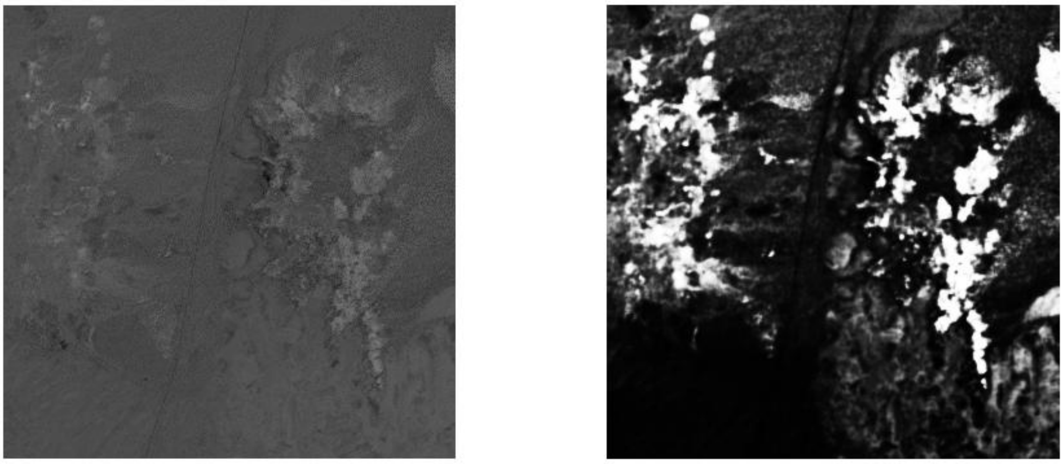
Going back to the equation (4.1) describing the proposed model, the threshold value, α, in the source term was chosen to be the mean in each band of the image. This threshold value updates dynamically at each iteration as new values of the image are updated, recalculating the value of α as a mean. It was done so because it will then help accommodate the fact that, naturally, some of the bands in the image are not correlated.

Thus we do not want to treat the bands equally. Otherwise, choosing one threshold value globally in the entire image would cause loss of valuable information in the other bands, for instance values below a particular α. So while information in one band might be high, represented as values close to 1, other bands contain low values and a "global" threshold variable will reduce accuracy of the model.

Now, the diffusion coefficient determines the intensity of the actual blur of the image by the model, a value of $\mu = 0.3$ was selected for most of the images tested after a number of different experiments were run. This value was based on each image. Different values can be assigned depending on the desired results as it depends how much blur the user wants to apply to the image, and also depends on the information that is contained in the image itself.

If this diffusion coefficient was set to be very low, say about 0.1 or less, then the amount of blurring or diffusion will not be as effective and there will not be much smoothing on the image. And if the diffusion coefficient was very high at around 1, then it can cause too much blur and loosing the accurate region boundaries and exact edge positions what is contained in the image. The figure(5.4) below shows comparison of the unprocessed Cuprite in figure(5.4a) and the processed Cuprite in figure (5.4b) images figure which took approximately 2 minutes to run the simulation.

We show below the artificial image, at the 180^{th} band where the alunite region is more prominent and all images presented here will be a representation of this particular band. This is the band choice to look at so as to compare with the results of C. Cox in [23]. Later, boundaries from the highlighted regions versus the original image will be shown to compare how the edge detection methods perform. Below we see the difference in regions detected as highlighted by the output compared to the original image on the left.

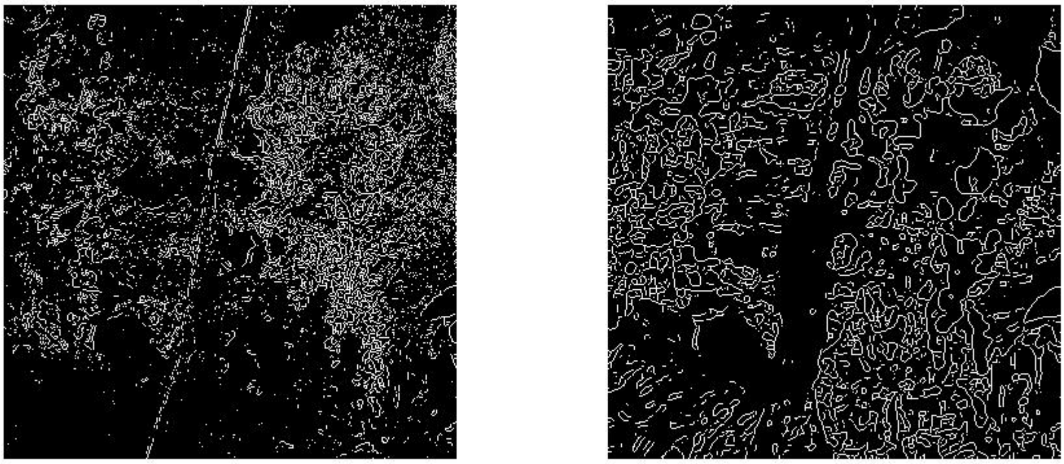


(a) (b)

Figure 5.4: (a) Is the 180^{th} band of the artificial cuprite image (b) shows the 180^{th} band after processing the Cuprite hyperspectral image with the proposed method after 7 iterations, $\mu = 0.3$, $\eta = 1$ and α as the mean described previously.

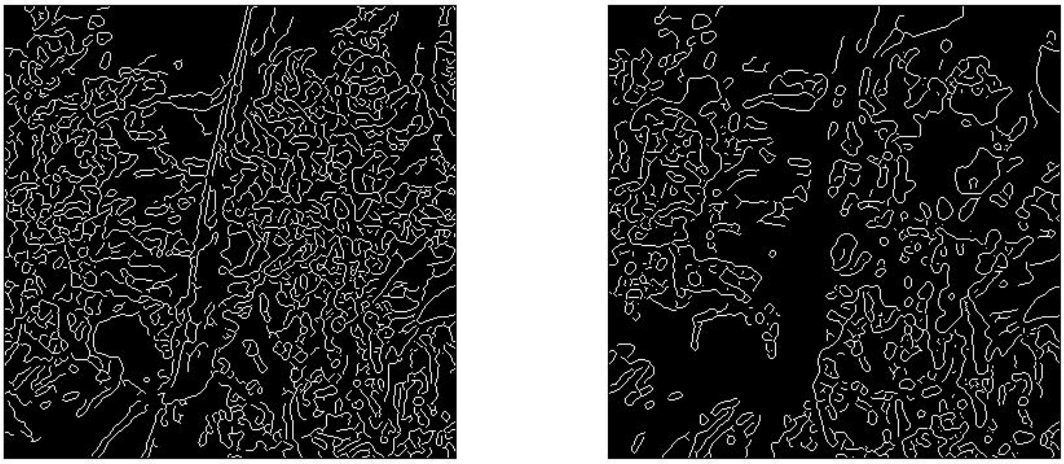
Regions within the Fitzhugh-Nagumo (FN) processed image are clearer, which is the purpose of the proposed method. This shows the portential of the FN method in hyperspectral image processing. A further analysis on the edge detection for the regions found by FN model compared to the regions in the original image is done below. We will further test and compare edge detection methods applied to the hyperspectral image data on unprocessed and those processed with the proposed FN method. The FN method was applied to the image as a whole and then comparison of edges is done band by band.

Now we look at the different results of the above images of the original unprocessed image and the FN processed image. In each of the images, we will compare at the 180^{th} band of FN unprocessed image and FN processed image. In the processed image, the proposed method was first applied on the image and then we did the comparison band by band. In other cases, we look at the principal components. This comparison is done on the Cuprite image and core image.



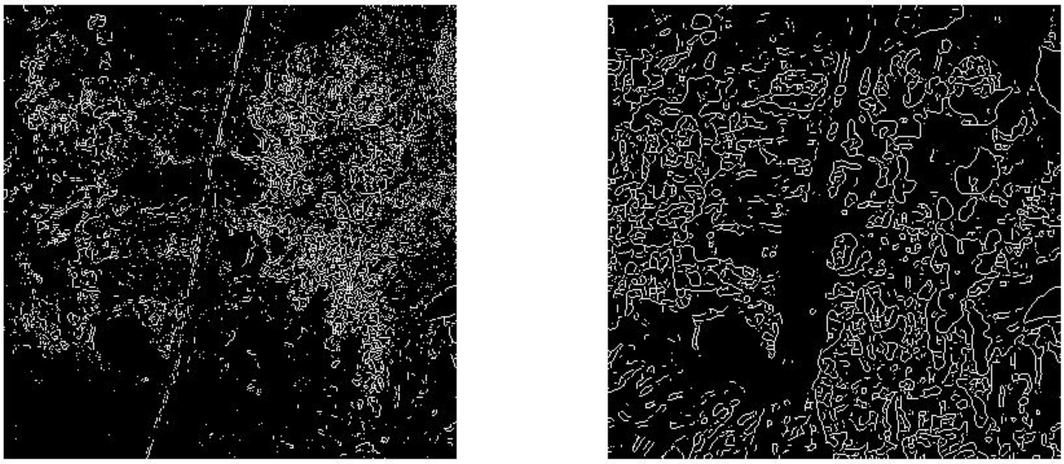
(a) (b)

Figure 5.5: (a) is the MatLab Prewitt edge detection output with threshold=0.002, (b) represents Matlab Prewitt edges with threshold=0.002 of 180^{th} band after applying the proposed FN method to the hyperspectral image.



(a) (b)

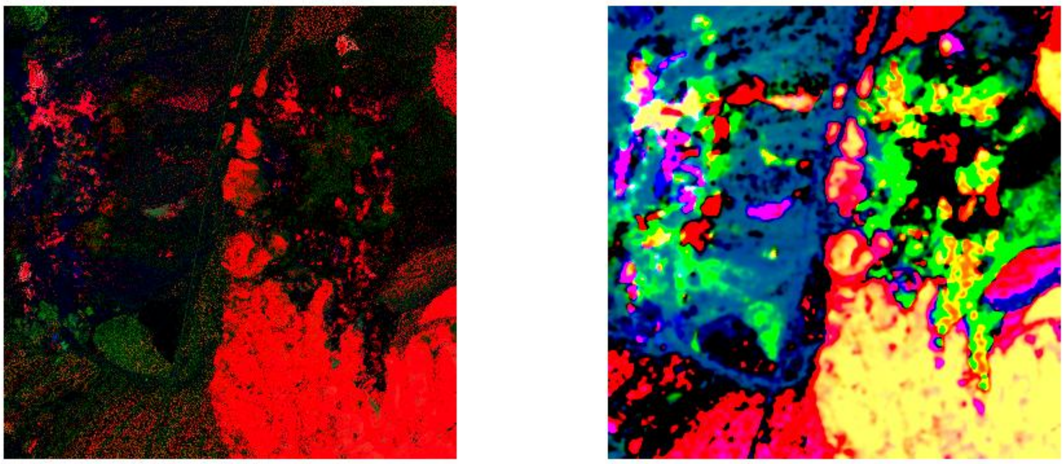
Figure 5.6: (a) Is the Canny edge detection with threshold=0.011 and sigma=0.012 on original image and (b) is the Canny edge detection on FN processed hyperspectral image.



(a) (b)

Figure 5.7: (a) is the Sobel edge detection output with threshold=0.002 on the same band as above of original image and (b) represents Sobel output with threshold=0.002 on FN applied image.

Here we see that of the first three PCA's, alunite is still dominant, appearing here as the green sections in the processed image on the right.

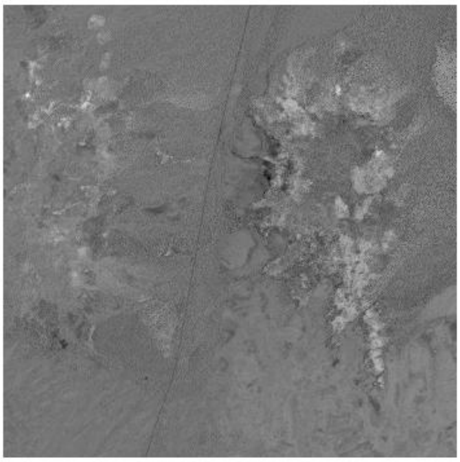


(a) (b)

Figure 5.8: (a) Colour composition first three principal components on the artificial image, (b) is the the first three principal components after applying the proposed FN method.

5.3 FN method applied to real hyperspectral core image

In this section, we now take a look at the application of the various methods to the real cuprite image. Similarly to the previous section, we will look into the edge detection outputs, PCA and in addition, we test the proposed method on added noise. Similar to figure (5.4), we see again the highlighted regions after applying the FN method which we can further analyse later for edges.

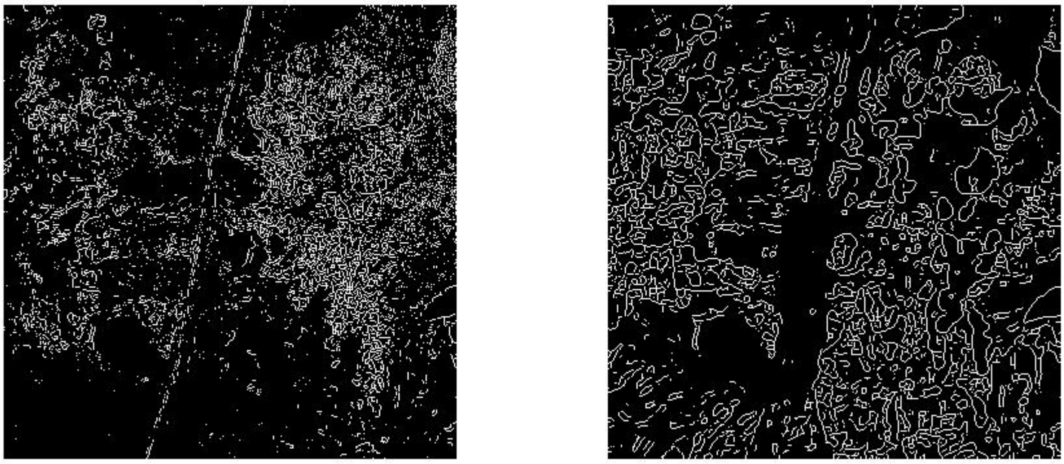


(a)

(b)

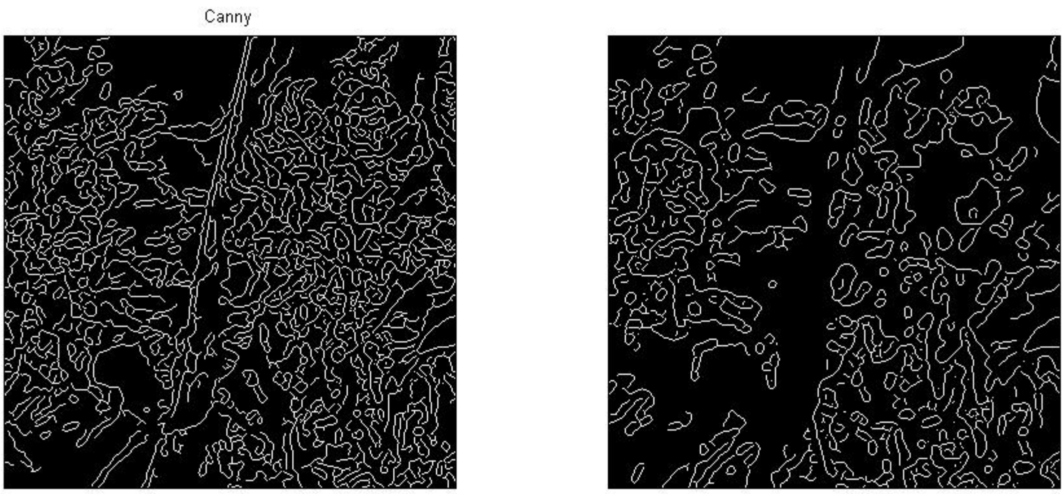
Figure 5.9: (a) Is the 180^{th} band of the real cuprite image (b) processed image by the proposed method after 7 iterations, with $\mu = 0.3$, $\eta = 1$ and $\alpha = 0.3$.

The processed image shows more blurred and smoothed regions when comparing to the initial images. Pixels values in the processed image in each band result from the nature of the PDE discretization [1]. For each pixel, the discretization of the model makes use of neighboring pixels for processing, as well as the boundary conditions. For the results obtained in this this research, the regions were well defined and highlighted which can be clearly seen in figure (5.9b) when comparing the original and the processed images. It is what we were hoping to achieve.



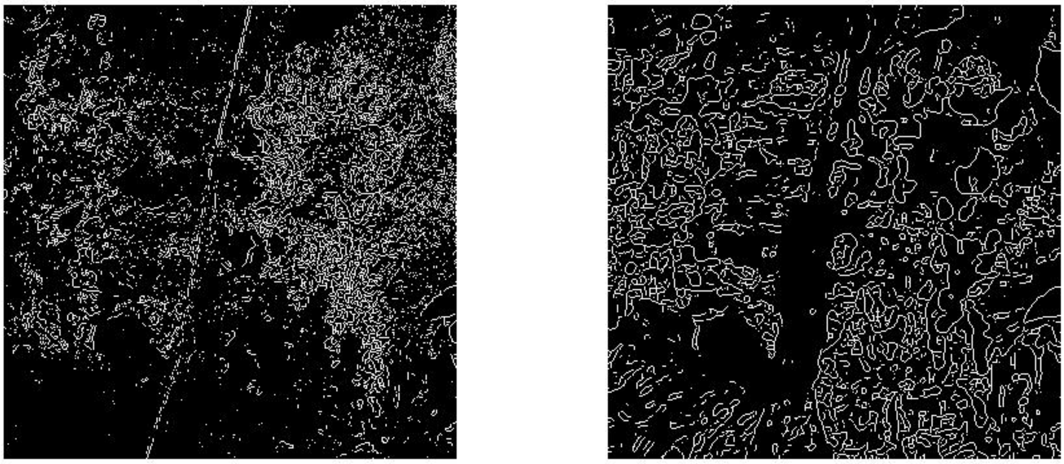
(a) (b)

Figure 5.10: (a) represents Prewitt edge detection output with threshold=0.002, (b) represents MatLab Prewitt edges with threshold=0.007 after processing with FN method.



(a) (b)

Figure 5.11: (a) represents Canny edge detection with threshold=0.011 and sigma=0.012 on original image, (b) Canny edge detection output with threshold=0.1 and sigma=1.5 on 180^{th} band of the FN processed method.



(a) (b)

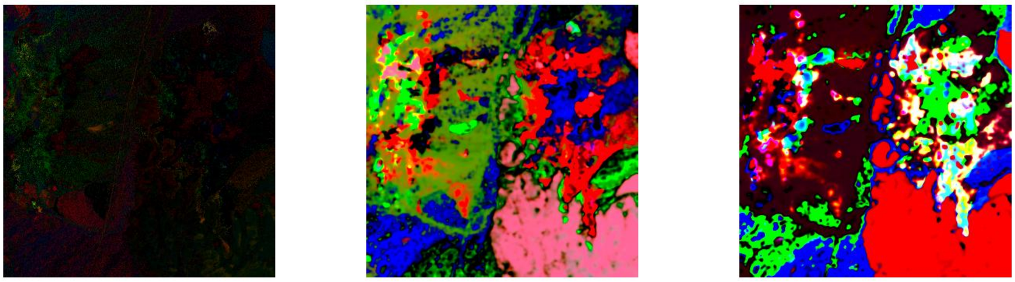
Figure 5.12: (a) represents Sobel edge detection with threshold=0.002, (b) is the Sobel edge detection output with threshold=0.002 after applying FN method.

We see that here the Prewitt and Sobel method on the proposed method shows the region edges much more clearly on FN applied images. Regions are better defining hence a clear indication of where the edges lie. Because of the diffusion effect in the proposed method, the road feature that goes from the bottom to the top across the center of the image has disappeared in all Canny, Sobel and Prewitt outputs. It is seen on the edge detection applied directly to the original image.

We further look at the results of the PCA. The figures reveal another property of the nonlinear Fitzhugh-Nagumo model when comparing the three images in figure (5.13) below. In terms of the brightness, figure (5.13b) (5.13c) appears brighter than figure (5.13a) with more information appearing in (5.13b) and (5.13c) from the dark regions in (5.13a). These colours are enhanced by the source term in the Fitzhugh-Nagumo equation as well as its threshold value. Pixel values in the image are moved away from the threshold value toward 0 and 1 giving more of the binary effect in these pixels as described in [1] . This result helps achieve more well-defined regions in the images.

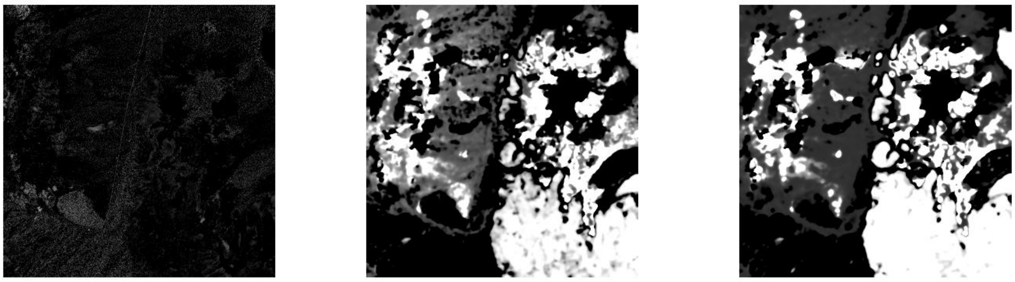
Figure (5.13) shows the PCA two to the fourth PCA of the image. PCA two, three and four is shown mainly because the first PCA shows a brightness of the image pixels. Figure 5.14, (5.15) and 5.16 shows PCA one, two and three respectively, while

(5.17) is the first PCA of brightness.



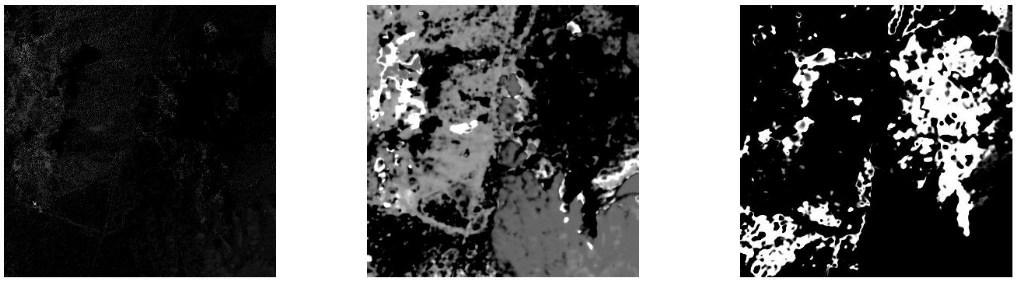
(a) (b) (c)

Figure 5.13: Principal components 2 to 4 of (a) Unprocessed artificial Cuprite data and the processed image (b) by the proposed model (after 7 iterations), (c) is obtained after 10 iterations.



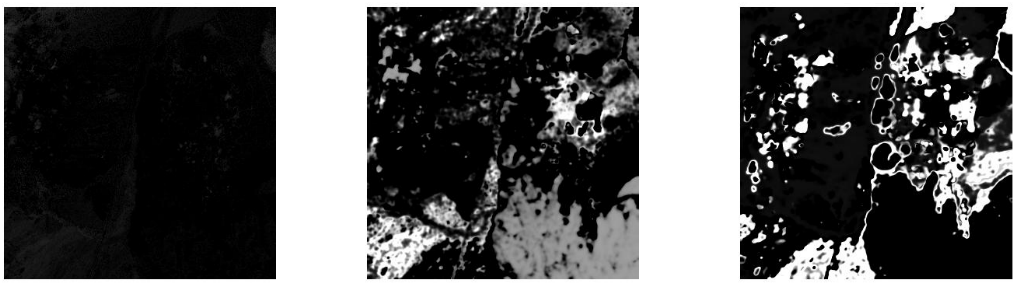
(a) (b) (c)

Figure 5.14: Principal components 2 of (a) Unprocessed real Cuprite data and the processed image (b) by the proposed model (after 7 iterations), (c) is obtained after 10 iterations.



(a) (b) (c)

Figure 5.15: Principal components 3 of (a) Unprocessed real Cuprite data and the processed image (b) by the proposed model (after 7 iterations), (c) is obtained after 10 iterations.



(a) (b) (c)

Figure 5.16: Principal components 4 of (a) Unprocessed real Cuprite data and the processed image (b) by the proposed model (after 7 iterations), (c) is obtained after 10 iterations.

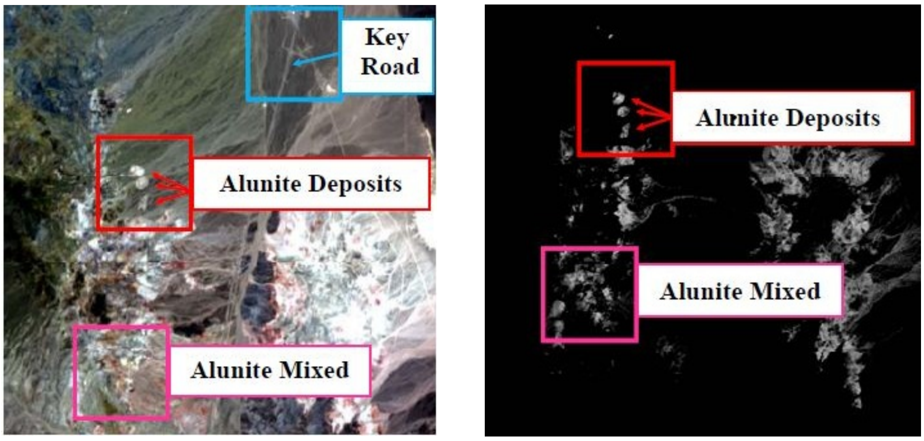
(a) (b) (c)

Figure 5.17: Principal components 1 of (a) Unprocessed real Cuprite data and the processed image (b) by the proposed model (after 7 iterations), (c) is obtained after 10 iterations.

Principal component analysis comes with its own advantages when comparing the processed and unprocessed image data. There are differences between the original image data and the processed data when they are both transformed using the principal component analysis. Since information in the processed data has changed compared to the original, due to the nature of the model, in this case information from different bands of material in the image changes the correlation. This affects the way the reduced dataset is represented using PCA form.

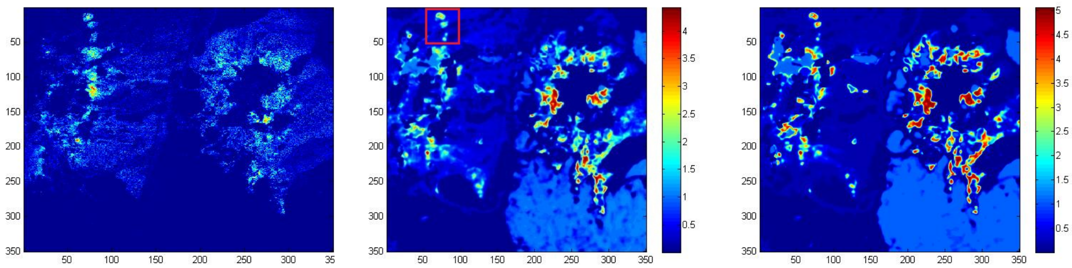
A comparison of the second highest principal component to the fourth highest of the images is done in figure (5.13). Depending on the type of images, the first principal component may represent the amount of brightness levels of pixels in the images. The colours in the figures are a visualization of the information from the entire image "compressed" into these RGB images. It is seen that the second principal component (red) part of the image appears almost in the entire bottom right of the image and this can represent material close to the alunite. The red regions mostly on the top left part of the processed figure are the important mineral information we are looking for - also alunite mineral in the original image which the endmember was obtained using the NFINDR, top right of figure (5.13b) and (5.13c) shows these regions much better than the figure (5.13a) of the image unprocessed with the FN method.

Now do a comparison of the Alunite dominant regions in the processed image and the original unprocessed image. This region was of particular interest in the Cuprite image from AVIRIS. We wanted to compare results obtained by the proposed method to those obtained previously by C. Cox in [23]. The colour scale in the next section of images indicate; red - where the FN method shifted values close to 1 showing that the region is highlighted out from the background and blue close to 0 where the FN method separates out the background so the highlighted region is much clearer.



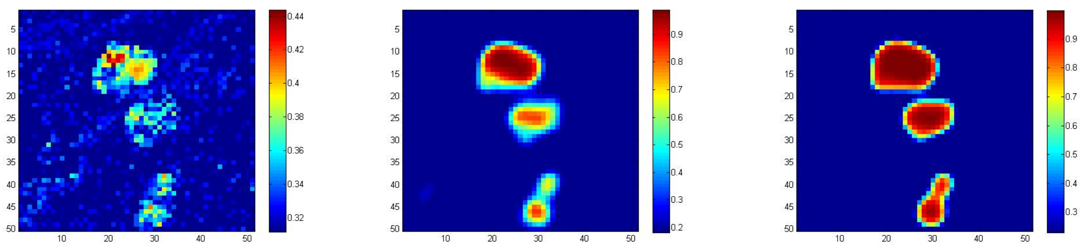
(a) (b)

Figure 5.18: Figure (a) is the cuprite image and figure (b) is the output from the HyS-PADE method (from [23]) using the spectral angle map technique.



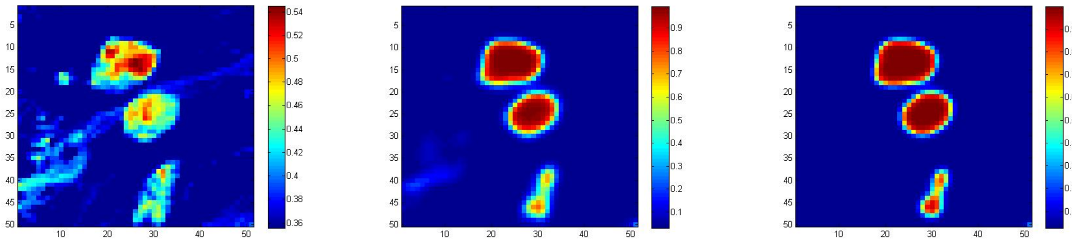
(a) . (b) (c)

Figure 5.19: (a) Alunite dominant image determined by Nfinder method, the processed image (b) after 7 iterations and (c) after 10 iterations with more defined regions taking the second major PCs in (b) and (c) - a close match to the Nfinder component.



(a) (b) (c)

Figure 5.20: The zoomed in (the red square) (a) Alunite dominant image band 180,(b) and (c) is the the same band after applying the FN method (with 7 and 10 iterations, respectively) with more defined regions. These are actual image bands without applying the principal component analysis.



(a) (b) (c)

Figure 5.21: The zoomed in (the red square) on the **real** Cuprite Nevada image (a) Alunite dominant image band 70 without impurities,(b) and (c) is the same band after applying the FN method (with 7 and 10 iterations, respectively) showing more defined regions taking. These are actual image bands without applying the principal component analysis.

Figure (5.21b) and (5.21c) show the degree of effectiveness in the model in comparison to the original (5.21a). Depending on the type of edge detection technique, edges revealed are defined as expected from the results. From the PCA point of view, which will be shown later, the method still works well in giving more defined regions. But, thresholding can become an issue here, and in general. The bottom region appears to shrink even though the distinct features are separated well from the background as seen in (5.21c).

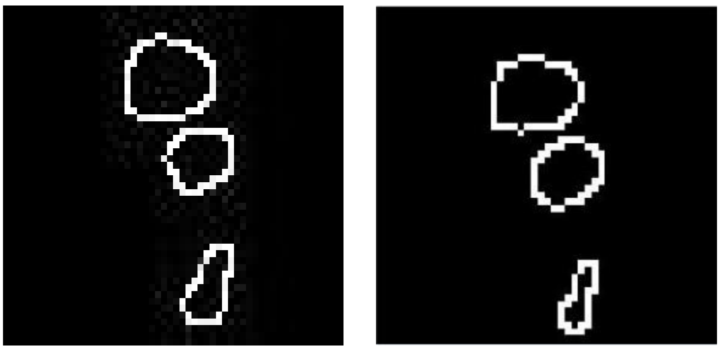
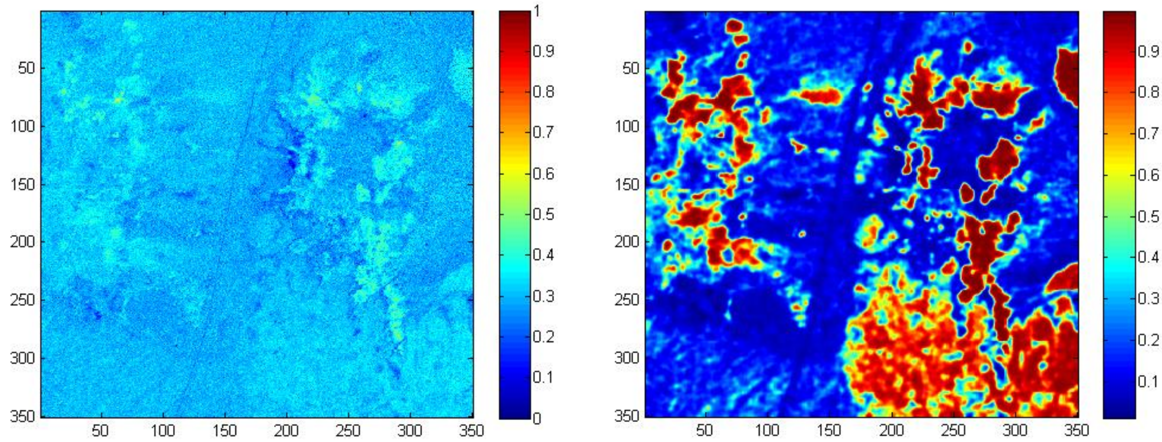


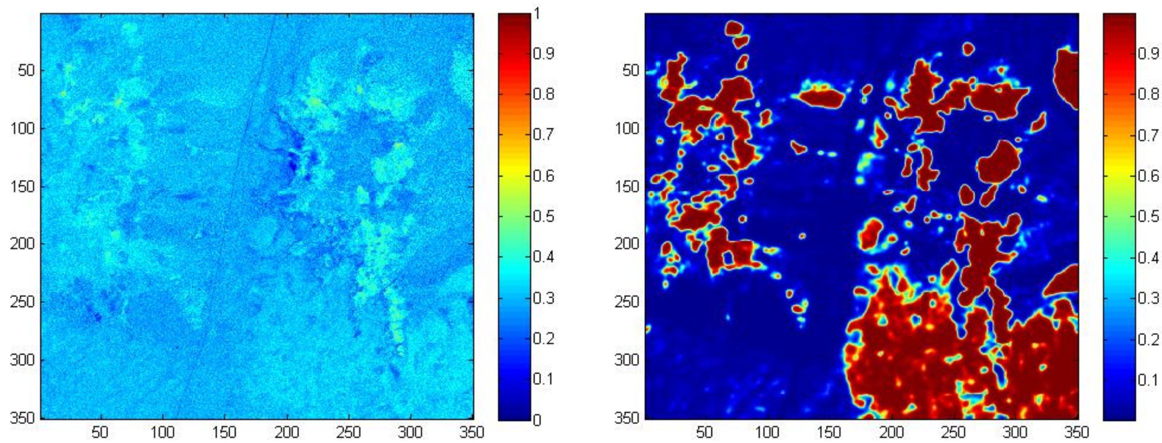
Figure 5.22: (a) Canny edges with a threshold value of 0.2. on the alunite dominant region. (b) Canny edges of image without impurities with a threshold value of 0.2.

In the following images, noise is introduced to the Cuprite image. Noise level was chosen here in such a way that you can still see the features of the image as in the original. MatLab packages of *imnoise* was used.



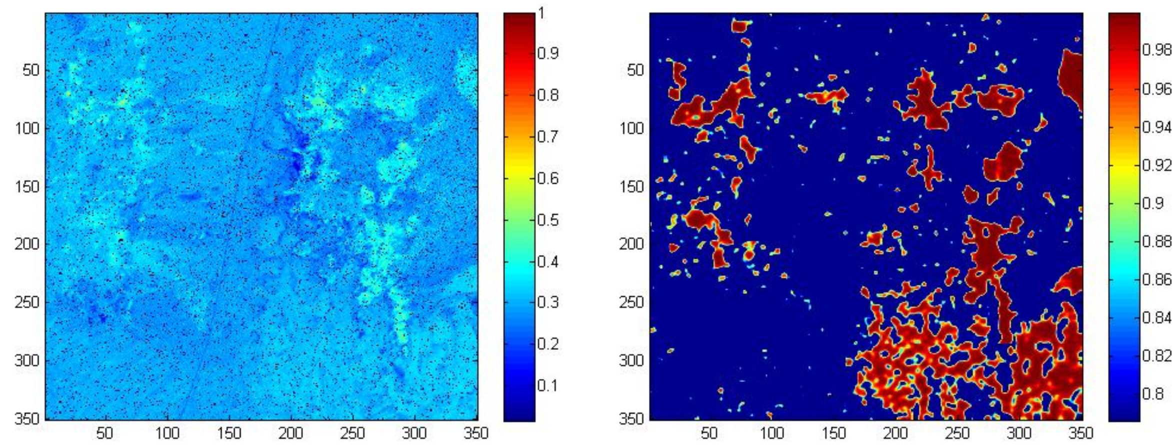
(a) . (b)

Figure 5.23: (a) Figure showing 180^{th} after after applying MatLab *Gaussian* noise with a mean of 0.002 and variance of 0.006, (b) is the the same band after applying the FN method (after 7 iterations) again show more regions as an effect of the noise.



(a) . (b)

Figure 5.24: (a) Figure showing band 180 after after applying MatLab *Speckle* noise at density of 0.005, (b) is the the same band after applying the FN method (after 10 iterations) again show more regions as an effect of the noise.



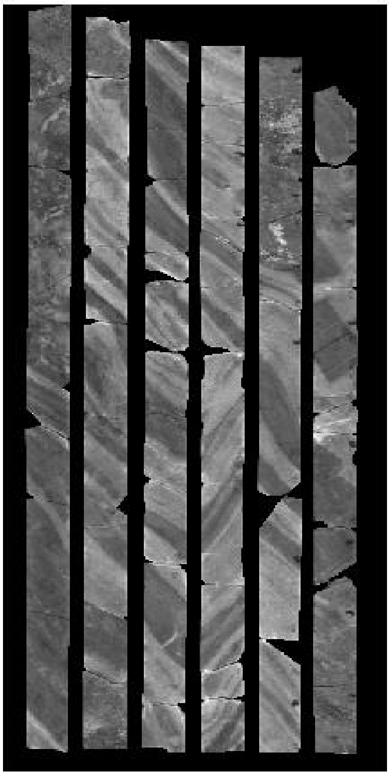
(a) . (b)

Figure 5.25: (a) Figure showing band 180 after large amounts of *Salt and Pepper* noise has been added at variance of 0.05, (b) is results of the FN method (after 10 iterations) again show more regions as an effect of the noise.

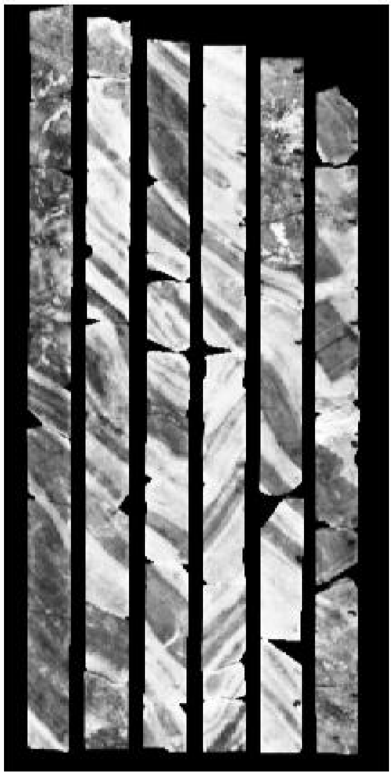
In the figures (5.23) - (5.25) we see how the proposed method works when noise has been added to the image in the form of (*Speckle*) and (*Salt and Pepper*). The method performs well when reasonable amounts of nose has been added in figure (5.23b) and (5.24). yet there is negative effect in the output of the image (5.25) when working with *Salt and Pepper*. Smaller regions in the output image occur but does not affect the overall shape of the regions. This shows that the method works well under natural hyperspectral Gaussian noise. However, when testing the limits of this method by adding a lot of noise which is unnatural, noting that *Salt and Pepper* noise is unusual for hyperspectral images. Regions of the output image change form as seen in figure(5.25) for the *Salt and Pepper* noise. This was done for testing purposes.

5.4 FN method applied to artificial core hyperspectral image

Below we look at the effect of applying the FN method on the artificial image compared to the artificial image without applying the FN method. Images on the left represent edge detection techniques applied directly to the 180^{th} band on the artificial image. On the right is the edge detection techniques applied to artificial images after the FN method is applied to it. First we have a look at band 180 of each image of the original artificial image and of that processed with FN. We see that as in the FN unprocessed image, regions remain but are more prominently visible in the FN processed image.



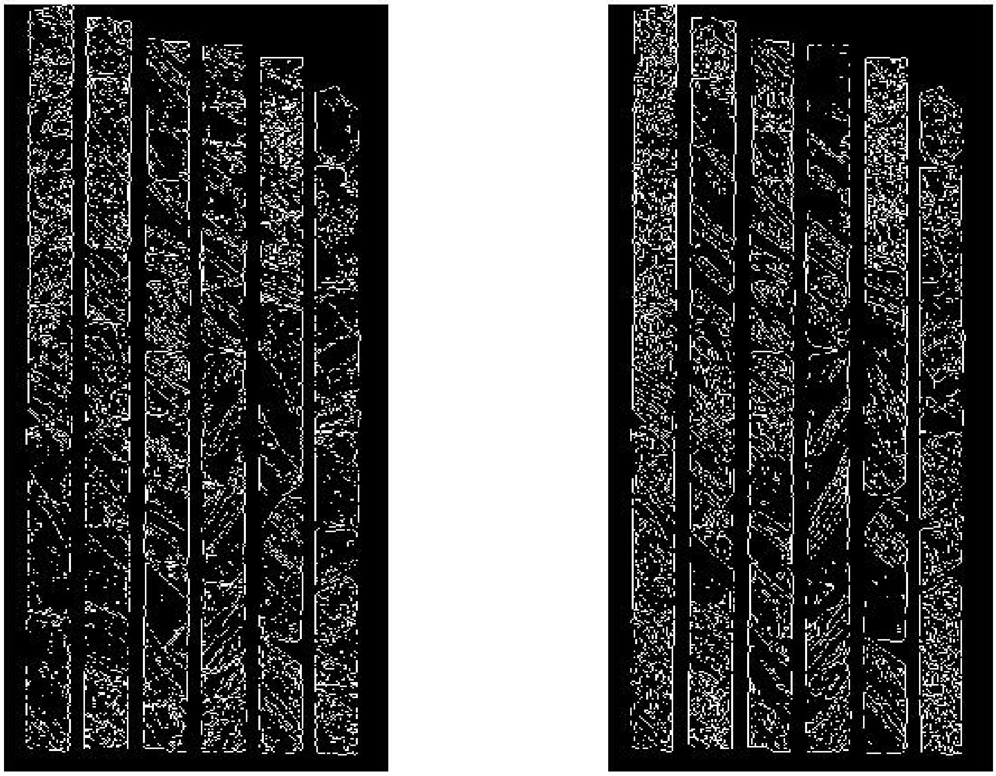
(a)



(b)

Figure 5.26: (a) Figure shows 180^{th} band of the artificial core data created by the process described in the methodology, (b) is the equivalent results on the FN applied image with 3 iterations.

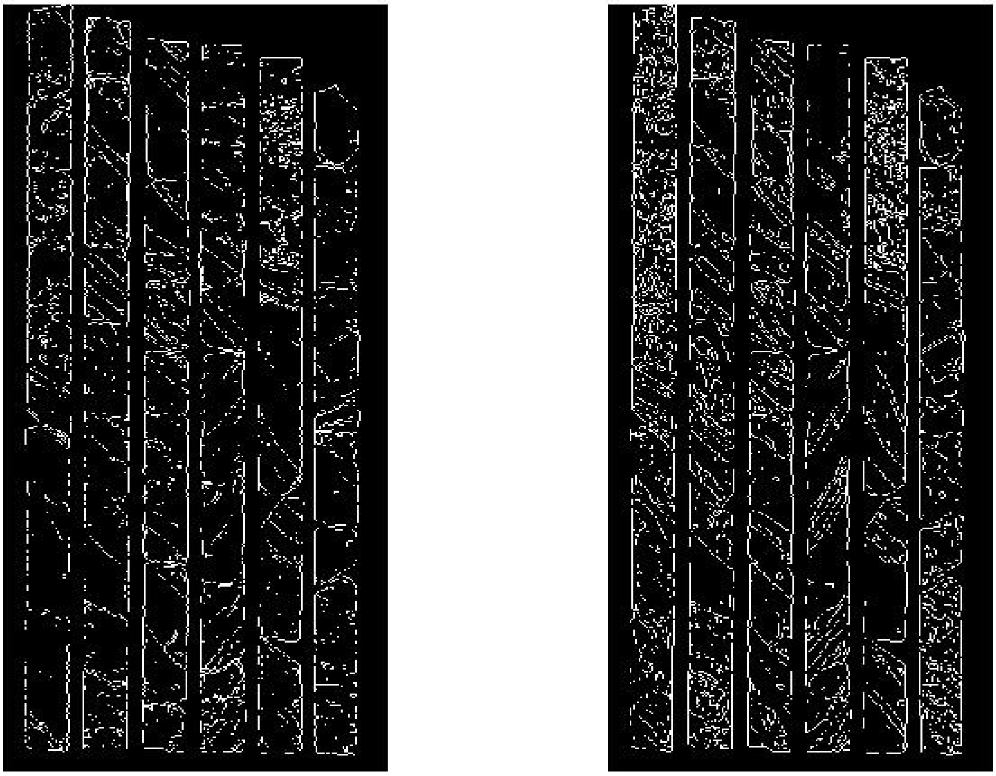
We now test the effect of of the proposed method when applying the edge detection method. We will test the Sobel, Prewitt as well as the Canny edge detection methods as done previously in the Cuprite test. This will also be done in the next section for real data.



(a) (b)

Figure 5.27: (a) Figure shows Sobel edges on the 180^{th} band of the artificial core data at threshold = 0.03, (b) is the equivalent results on the FN applied image with 3 iterations.

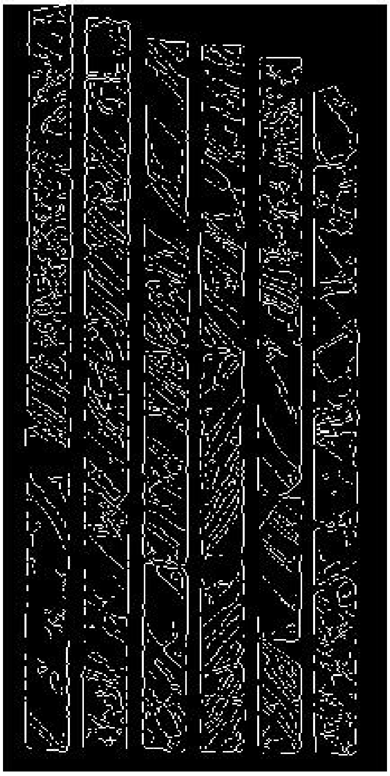
Output on the Sobel shows that there is no obvious difference in the comparison of the two images. Upon observing the core structures carefully, we see that the edge detection technique found more edges in the FN processed image, for example the top part of the top of the first core structure from the right.



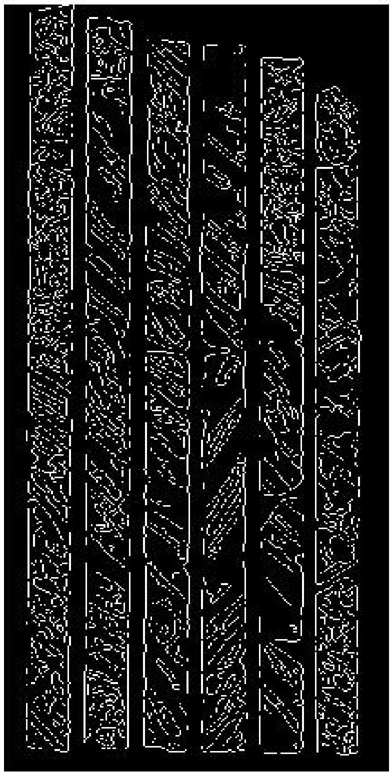
(a) (b)

Figure 5.28: (a) Figure shows Prewitt edges on the 180^{th} band of the artificial core data at threshold = 0.05, (b) is the equivalent results on the FN applied image with 3 iterations.

Prewitt technique shows more defined structures within the cores themselves than the Sobel method. This time on the FN applied images, more edges are captured as compared to the artificial images on the left where FN method was not applied. This can be seen on the third cores from the left.



(a)

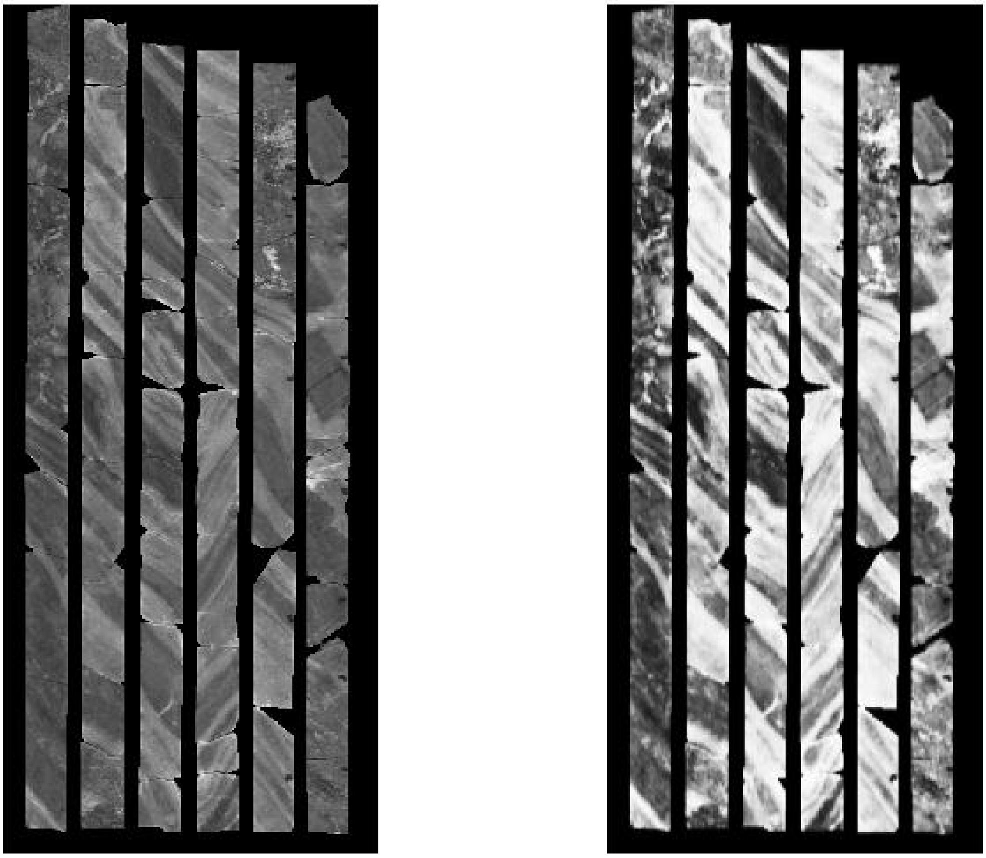


(b)

Figure 5.29: (a) Figure shows Canny edges on the 180^{th} band of the artificial core data at threshold = 0.05, (b) is the equivalent results on the FN applied image with 3 iterations.

We see that with the Canny edge detection method, the output shows well defined edges in both of the images. The difference here come on the first and the last cores in terms of the difference in edges captured.

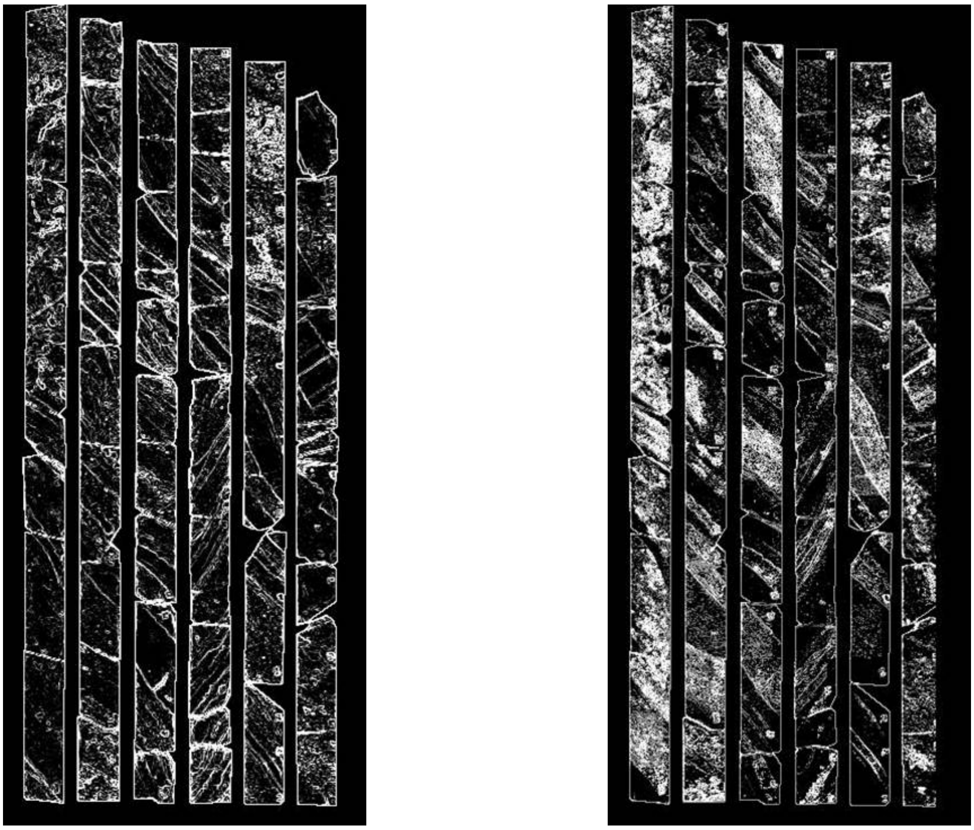
5.5 FN method applied to real core hyperspectral image



(a) (b)

Figure 5.30: (a) Figure showing 180^{th} band of the original core image, (b) equivalent image on FN applied image (after 7 iterations with diffusion coefficient 0.2).

The core image has a much higher resolution with much smaller pixel compared to the Cuprite image. In the above drill core images, the enhancements on the first drill core are interesting to look at in that we see more detailed features across its entire length. The model managed to enhance the features here, such as the triangular shape near the bottom left part of the image. The mineral dominant in this part of the image seems to be spread across the entire image.



(a) (b)

Figure 5.31: (a) Shows the Robert's technique result (courtesy of R. Resmini)(b) Shows the first 11 PCA of HySPADE technique output (courtesy of R. Resmini).

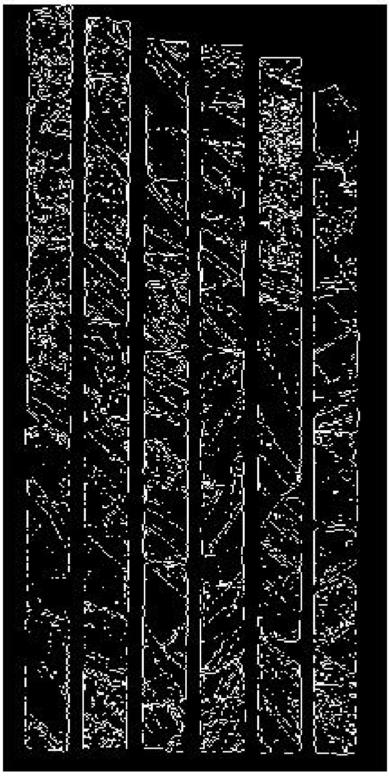
The results in the figure (5.31) were ran separately by Ronald Resmini, author of the HySPADE method [20]. When compared to the Robert's method, the HySPADE result appears better. In the figure, it can be clearly seen that some of the highlights in the image appear, thus it captures most of the edges as well as the regions themselves. We also do the comparison with the proposed method below. In the proposed technique, we see that only the edges around the regions a are shown due to the nature of the model.

(a) (b)

Figure 5.32: (a) Figure shows Canny edges on the 180^{th} band of the core data, (b) is the equivalent results on the FN applied image with 7 iterations.

(a) (b)

Figure 5.33: (a) Figure shoes Sobel edges on the 180^{th} band of the core data, (b) is the equivalent results on the FN applied image with 7 iterations.



(a)

Figure 5.34: (a) Figure shoes Prewitt edges on the 180^{th} band of the core data, (b) is the equivalent results on the FN applied image with 7 iterations.

In the core image, the proposed method again produces smoother edges than the counterpart techniques although with the kernel used some of the edges appear finer than others but yet are still visible. Most of the Canny edges are broken and less continuous but are clear and much more visible. The Sobel and Prewitt techniques mostly found boundary edges rather edges within the core. These are not continuous edges.

Figure 5.35: Figure representing first core strip from the left processed with the FN method with $\mu = 0.001$ and $\eta = 1$ iterated 3 times.

Figure 5.36: Image represents the Canny edge detection output on original image with 0.1 threshold and 1.3 standard deviation.

Figure 5.37: Image represents the Canny edge detection output on FN processed image with 0.1 threshold and 1.3 standard deviation.

(a) (b) (c)

Figure 5.38: Figure showing the representation of the first three principal components of (a) the original core data ,(b) result from [19] and (c) the proposed model. In this images, red represents the first Principal Component, green the second, blue the third and combination of the colours indicate a corresponding combinations of the components that are present.

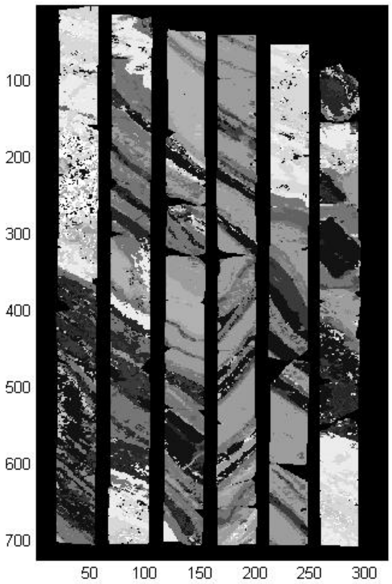
(a) (b) (c)

Figure 5.39: Canny edge detection where (a) is a direct application of the threshold to original first principal components (b) is the results by [30] and (c) proposed model result.

(a) (b)

Figure 5.40: (a) Principle component 2 - 4 of real data, (b) Principle component 2 - 4 of real data after processing with FN method. The colour representations identify the Principal Components where red is the second principal component, green is the third, blue is the four and combination of colour represents combination of the components.

The following image is the gray scale image of the real core data and the result of the SOM method used at the Geospectral Imaging (Pty) Ltd.



(a) SOM method results, from the company Geospectral Imaging (Pty) Ltd.

(b) Proposed method after 10 iterations on the real image.

5.6 Summary

In summary, the proposed method works well in the 3 band RGB image of the coin. Clearer edges resulted from the application of the FN method as the features of the coin were enhanced while removing noise, as seen on the background comparing the Canny edge detection results.
In the Cuprite image, it was clear to see the alunite regions in the comparison of the images where FN method was applied and images where the FN method was not applied. In most of the image, there where much clearer edges on the FN results than on the original results as we had much better identification of regions by the FN method. However, applying the proposed FN technique removed some of the features such as the road, hence it was not clear in the FN outputs. This can also be seen in the results of the HySPADE method.
Analyzing the proposed method on images with different noise levels also enable testing the smoothing limits while retaining structural features in the image. Gaussian noise is common in hyperspectral images. Gaussian noise was added for testing

purposes and the results show that FN technique removed noise effectively and enhanced regions of the image. Similarly for the Speckle noise. Output on the salt and pepper noise causes deformation of the features and this marks the limit on the technique.

Core results of the HySPADE show very clear edges of the individual core edges as well as on the body on the core. Outputs on the edge detection of the FN method performed similarly to applying each of the edge detection techniques directly on the original image.When looking at an individual core strip, and applying the FN method, we see that FN method shows the features on the core much more clearly resulting in the better edges of the FN applied image.

Chapter 6

Conclusions

6.1 Conclusion on the proposed 3-dimensional Fitzhugh-Nagumo model compared against SOM

In this study, the development of the nonlinear diffusion equation of the Fitzhugh - Nagumo method and its application in image processing was introduced. The 1- and the 2-dimensional cases were introduced in [1] amongst other papers with a detailed application on the Heaviside function and on text binarization. It was shown that its diffusive properties are also applicable in the 3-dimensional dataset of hyperspectral images.

The diffusivity coefficient, μ, as well as the scaling factor, η, of the source term in the model equations have very important outcomes in the processed images. First, μ is a factor that determines the amount of blurring in the image to apply at each time. It is important to note that not only does the blur occur in the spacial dimensions, but it also occurs in the spectral dimension, the z-direction in proposed FN model equation. This causes the "energy" from neighbouring bands to have an effect on the other bands. So the energy in band $i-1$ and in band $i+1$ of the same pixel will have an effect on the energy of band i of that pixel. This diffusion occurs in the same way as it does in the pixels' spatial x- and y-directions. This can have both positive and negative impacts on the output image. We may have that the spectra of some minerals occur in certain patterns and once the diffusion process is done in the image, the patterns may change causing some unwanted patterns when observing the results from the spectral point of view. That is, it can change the spectral signature of the minerals or endmembers contained within the image, since diffusion in the x- and y-directions was also applied between iterations, plus the values in the image matrix were forced to be near 0 or 1 according to the threshold source term in the proposed model. Hence this method is ideal to visualize the results spatially, in the normal x- and the y-dimensions where we are able to visualize the separation of the different endmembers.

The proposed method can be used for different types of images with multiple bands such as the normal Red-Green-Blue (RGB) colour-combination image from different types of cameras and can also be used on hyperspectral images from specialised hyperspactral cameras as illustrated in this study. In RGB images, depending on how the user wants to process the images for either diffusion or highlighting features in the image, it can be concluded that the number of bands can affect the number of iterations. For instance, in the case of the coin, figure (5.2), only 3 iterations were required to reduce noise significantly and retain features of the image, while for the hyperspectral images, the number of iterations was higher between 7 to 10 iterations.

In the comparison of the proposed method against SOM, it is seen that the developed method shows highlighted regions that contain minerals in both the Core and the Cuprite image. This is done with the aid of the source term of the model. Thus if two different minerals show more or less the same amount of energy in a band, then the proposed method will be able to detect the differences with the background but not the differences between the two. However, the SOM results show that the method can separate where the different minerals occur but can also show finer details like noise, whereas the proposed method smooths out the details and hence less detail will be detected while keeping the overall shape of the regions.

The proposed method does well in effectively removing the noise in the image as seen in the coin example figure (5.2). But this comes with the effect of loosing some of the information in the image and hence would not work well on all images types, especially images with information with finer detail similar to noise. At the same time, the source term helps to separate out pixels that are significantly different to each other or to the background. Hence, the proposed method here works out to “binarize” hyperspectral images at each of the image bands.

6.2 Conclusion on the proposed 3-dimensional Fitzhugh-Nagumo model compared against HySPADE method

As introduced in the literature review, the HySPADE model from the paper by R. Rasmini [9] makes simultaneous use of the spatial and the spectral information of the hyperspectral images. The implementation of this method was originally done in Visual studio, using the C programming language. For this research, the implementation was initially done in the CodeBlocks interactive development environment (IDE) which is a free software (downloaded from [31]) that provides the similar C programming language. This output result for the HySPADE method was difficult to visualize in the ENVI software due to technical issues.

In order to get results to compare these methods, the code was consequently redone in MatLab. This is because once the image is loaded into MatLab, it can easily be visualized using functions such as the *imshow* and *imagesc*. Now, the disadvantage of using the HySPADE technique in MatLab is that it runs much slower, and thus take a lot of time. In the CodeBlocks IDE, the HySPADE method runs up to 30 minutes for the core imager data and up to 15 minutes for the Cuprite image, while the method runs for up to 26 hours in MatLab, to process the same images. The runtime in the original paper [9] was said to take several minutes and this method makes use of large number of computations.

HySPADE presents very good results in that the edges within and around the drill core are well preserved and most of the detail about the image also still remains. Compared to results of the proposed method, regions in both techniques are well identified. The proposed FN method shows mainly boundary of each of the features within the core structures as most of the detail has been diffused out and features highlighted as seen in figure (5.33b) with $\mu = 0.2$, $\eta = 1$ over seven iterations. We see also that the proposed method can process well with detail over a single core structure as seen in figure (5.35). Figure (5.31b) output is the first 11 PCAs's of the HySPADE processed with a 50x50 window size.

6.3 Conclusion on the Prewitt, Sobel and Canny Edge

Image results of edges obtained for the Sobel, Prewitt and Canny are obtained using MatLab built-in *edge*. Default values of each of the functions were used for the Prewitt and Sobel techniques. These methods detected main edges between each core and the core box, which reveals the outer boundaries of each drill core. Canny edges appeared to be much better than Prewitt and Sobel and much more comparable with the proposed method. Canny edge is much more detailed compared to the edges obtained after applying the proposed method. What the proposed FN method did further is to separate out the different regions in the core and show much more clearly where regions are and less of multiple edges.
These types of data are ideal to test the proposed FN method as they exhibit much more detail than testing on simpler images such as the coin on a white surface.

6.4 Final remarks

It is seen that the Fitzhugh-Nagumo model has quite an extensive application in the study of applied mathematics as briefly described in the introduction as well as the application in this research. One study here in hyperspectral image processing is identified. We have seen the advantages and the disadvantages of the proposed model in terms of image processing. There are various ways of choosing a threshold to outline regions and to see how effective the diffusion process should be. Below is the summary on the proposed model:

- choosing the threshold value: a way in which the threshold is chosen depends on the nature of the image and quality, as issues such as noise can have an effect on the output. Also depends on the type of noise and how much of it is present. The second issue around choosing a threshold is that each time the process is run, the user has to input a threshold value until a desired result is reached.

- there are various other schemes that can be used to discretize the partial differential equation in 3-dimensions as studied here. One of the methods that can be used to discretize the PDE is the Douglas-Gunn method [26] which is stated to be unconditionally stable, even for the 3-dimensional equation. However, as a further study, it can be researched if this is true for the nonlinear variant of the diffusion equation used in this research.

- an additional point that is a potential study to further this research is the simultanious region detection as well as edge detection instead of these being done seperately and finally,

- this model can also be applied to events such as diffusion of gas in space with a different form of source as this is a 3-dimensional diffusion model. So instead of the third dimension being different bands in a hyperspectral image, it would represent a third spatial dimension which is depth in space.

Bibliography

[1] B. Jacobs, E. Momoniat, *A novel approach to text binarization via a diffusion based model*, Applied Mathematics and Computation, 2013:446-460.

[2] `http://www.microimages.com/documentation/Tutorials/hyprspec.pdf`, accessed in February 2014.

[3] `http://simpleclimate.wordpress.com/2010/11/06/prehistoric-co2-double-up-gives-\warming-data`, accessed in March 2014.

[4] P. Harris, K. Winkler, M. Sears, *Hyperspectral Imaging of Drill Core? How and Why?* Presented at the Australian Earth Sciences Convention, Melbourne, Australia, 2006.

[5] M. Lee, L. Bruce, *Applying cellular automata to hyperspectral edge detection*, IEEE, Graduate student report, Mississippi State University, Mississippi State, MS 39762, 2010.

[6] A. Mahmood, A. Robin, M. Sears, *Modified Residual Method for the Estimation of Noise in Hyperspectral Images*, IEEE Transactions on Geoscience and Remote Sensing (to appear).

[7] V. Duin, R. Leitner, P. Paclik, *A method for edge detection in hyperspectral images based on gradient clustering*, research report, Delft University of Technology, 2009.

[8] S .Verzakov, P. Paclik, R. Duin, *Edge detection in hyperspectral imaging: multivariate statistical approaches*, Delft university of Technology, Information and communications theory group, The Netherlands, 2006

[9] R. Resmini, *Simultaneous Spectral/Spatial detection of Edged for Hyperspectral Imagery: The HySPADE Algorithm Revisited*, Report for the MITRE Corporation, 2012.

[10] J. Ansamaki, J. Parkkinen, P. Toivanen, *Edge detection in multispectral images using the self-organizing map*, Pattern Recognition Letters 24, 2003:2987-2994.

[11] J. Canny, *A Computational approach to edge detection*, IEEE Transactions on Pattern analysis and Machine Intelligence, Vol. PAMI-8. No. 6, 1986

[12] P. Paclik, R. Duin, G.M.P. van Kempen,and Kohlus R, *Segmentation of multi-spectral images using the combined classifier approach*, Image and Vision Computing Journal 21, 2005:473-482.

[13] S. Di Zenzo, *A note on the gradient of a multiimage*, Computer Vision, Graphics, and Image Processing 33, 1986:116-125.

[14] T.L. Huntsberger and M.F. Descalzi, *Color edge detection*, Pattern Recognition Letters 3, 1985: 205-209.

[15] M. De Biasio, R. Leitner, F. G. Wuertz, S. Verzakov, P. Elbischger, *Enhancement of m-fish images using spectral unmixing*, International Journal of Biomedical Sciences, 2005:116-123.

[16] T. Kohonen, *Self-organizing map*, Neurocomputing 21, 1998:1-6.

[17] B. Mandelbrot, *Fractals: Form, Chance and Dimension*, W. Freeman and Company, 1977.

[18] R. Stevens, A. Lehar, F. Preston, *Manipulation and presentation of multidimensional image data using the Peano scan*, IEEE Trans. Pattern Anal. Machine Intell 5, 1993:520-526.

[19] P. Padayachee, Honours Research Report, Computer Science, University of the Witwatersrand, 2013.

[20] R. Resmini, *Hyperspectral/Spatial Detection of Edges (HySPADE): An algorithm for spatial and spectral analysis of hyperspectral information,* Proceedings of the SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery X, S.S. Shen and P.E. Lewis, eds., Orlando, Fla., April 12-16, v. 5429, doi: 10.1117/12.541877, 2004:433-442.

[21] G. Anusha, T. Prasad ,D. Narayana *Implementation of SOBEL Edge Detection on FPGA*, Electronics and Communications department, J.N.T.U, International Journal of Computer Trends and Technology, vol 3, Issue 3, 2012.

[22] D. Mumford, J Shah, *Boundary determination by minimizing functionals, I*, Proceedings on Computer vision and Pattern recognition, San Francisco, California, 1985.

[23] C. Cox, *Hyperspectral/Spatial Detection of Edges (HySPADE) Analysis of Mineral Assemblages and Linear Features at Cuprite, Nevada, U.S.A,* Department of Earth Systems and GeoInformation Sciences, George Mason University, Fairfax, VA 22030, ASPRS/ MAPPS 2009 Conference, 2009.

[24] K. Cawse-Nicholson, M. Sears, A. Robin, *Evaluation of bands containing spectrally correlated noise in hyperspectral imagery,* presented at IEEE WHISPERS, Gainsville, Fa USA, 2013.

[25] K. Cawse, M. Sears, A. Robin, S. Damelin, K. Wessels, F. van der Bergh, *Using random matrix theory to determine the number of endmembers in a hyperspectral image*, Whispers, Reykjavik, Iceland, 2010.

[26] J. Thomas, *Numerical Partial differential equations, finite difference methods*, Springer Text in applied mathematics 22, 1995.

[27] `http://aviris.jpl.nasa.gov/data/free_data.html`, (visited on April 2015).

[28] R. Roger, *Principal components transform with simple, automatic noise adjustment*, Int. J. Remote Sensing, vol. 17, no. 14, 1996:27192727.

[29] I. Jolliffe, *Principal component analysis*, second edition, Springer series in statistics, 2002.

[30] M. Longay, *Shape Detection in Hyperspectral Drill Core Images*, Honours Research Report, Computer Science, University of the Witwatersrand, 2013.

[31] `http://www.codeblocks.org/downloads/26`, (visited on May 2014).