Image processing, filtering and segmentation of data-sets for reservoir simulation

PhD Thesis

Tiril Pedersen Gurholt

Department of Mathematics
University of Bergen

Norway

June 2013

Acknowledgements

First and foremost I would like to thank my supervisors Professor Ivar Aavatsmark at Centre for Integrated Petroleum Research (CIPR), Professor Antonella Munthe-Kaas at the University of Bergen and Professor Arne Skauge at CIPR for their guidance.

There are many that deserve a thank you for their support during these years. A few people do, however, deserve to be mentioned. I am sincerely grateful to my current and previous colleagues at CIPR/University of Bergen, especially Roland Kaufmann, Bartek Vik, Andreas Sandvin and Eirik Keilegavlen, for the many interesting discussions and the support they have given me.

Tiril Pedersen Gurholt
Oslo, June 2013

Contents

B Determination of Connectivity in Vuggy Carbonate Rock Using Image Segmentation Techniques

C Pore space characterization of vuggy carbonate rocks: A comparative study of the performance of various image segmentation techniques

D 3D Multiphase Piecewise Constant Level Set Method Based on Graph Cut Minimization

Preface

This dissertation is submitted as a partial fulfillment of the requirements for the degree Doctor of Philosophy (PhD) at the University of Bergen (UiB). The working environment was *Centre for Integrated Petroleum Research* (CIPR) in collaboration with the Department of mathematics at the University of Bergen, Norway. The thesis was funded by CIPR.

Image data occur within different fields of petroleum research. They can be a result of laboratory experiments or geology field trips, and often result in a vast amount of image data that needs to be analyzed. Depending on the quality and size of the data they can be difficult to analyze. Within petroleum research the larger overall goal is improved knowledge of oil mobilization and reservoir characterization.

In this thesis various image processing techniques are studied. Two types of image data are encountered that are processed using two different types of techniques. This is reflected in the layout of this thesis, which is split into two theoretical parts. The first topic is filtering of X-ray image time-series coming from fluid flow experiments conducted at CIPR. These data-sets have image artifacts that can occlude the depicted fluid flow and makes it more difficult analyze the data. The goal of this work is to improve the presentation of the data. Good results are obtained that largely improves the presentation of the data.

Filtering is often a precursor to image segmentation, which is the topic covered in the second part of this thesis. There different types of segmentation methods are studied with respect to segmentation of computed tomography (CT) images of vuggy carbonate rocks. The goal of this part of the thesis is to determine which segmentation method is best suited for segmenting the image data-sets. The methods performance are evaluated for two-dimensional (2D) and three-dimensional (3D) images. It is generally seen that the image segmentation methods perform well on high resolution image data-sets. For decreasing image resolution the accuracy in the segmentation also decreases and there is more variations in the results.

Outline

This thesis is organized into three parts. In Part I the objective is to improve the presentation of 2D image time-series of fluid flow experiments acquired using an in-house X-ray scanner. This scanner introduces severe horizontal and vertical line artifacts which partly occlude the depicted fluid flow. These artifacts can be attenuated in the frequency domain after constructing an appropriate filter. In this part an introduction to frequency domain filtering and the filter construction is

given, before presenting the results and giving a conclusion. In Part II the objective is to study the performance of various image segmentation techniques for segmentation of CT images of vuggy carbonate rocks. These image volumes have varying image resolution and therefore also varying degree of difficulty. In this part an introduction to the segmentation techniques are given, before presenting the results and giving a conclusion. In Part III the included papers are presented.

Contribution of this thesis

Part I In the first part of this thesis image filtering of laboratory X-ray image data-sets of fluid flow experiments is considered. These image time-series exhibits periodic and a periodic image artifacts that can be attenuated naturally by frequency domain filtering after constructing an advanced filter. This filter is given as a combination of several different frequency domain filters. The developed procedure semi-automatically detects the artifacts frequency components and automatically computes the filter. The user need only provide the type of filters to be included. The artifacts remains fixed throughout an image time-series. It is therefore sufficient to compute the frequency domain filter once for each data-sets. The developed procedure has been published as a part of Paper A. It was later been modified slightly to further improve the results.

Part II In the second part of this thesis 3D segmentation of CT imaged vuggy carbonate rocks is considered. There a comparison of the performance of five different methods is given, namely standard global thresholding, the *Ridler and Calvard's* (RC) method [25], *Otsu's* method [23], local thresholding by *indicator kriging* (IK) [20] and region based segmentation using the *piecewise constant level set* (PCLS) method [15]. To validate the performance of the segmentation methods the expected visible porosity of the data-sets is computed. The detected porosity, pore length and pore volume distributions are also computed from the segmentation. This comparative study was first published in Paper B for three of the segmentation methods and it was later extended by the inclusion of two additional methods in Paper C (submitted June, 2013). To apply IK to our data-sets the original formulation was slightly modified.

In addition, a 3D graph cut solver was developed for solving the PCLS method. Traditionally, level set methods are solved by deriving and solving an *Euler-Lagrange* equation, which is time consuming. In this work it was shown that a faster solver is obtained by representing the problem on a spacial graph and computing the *minimum cut* on this graph. The presented graph cut solver is an extension of the 2D graph cut algorithm proposed in [2]. This resulted in the development of a fast and accurate solver, that is not very sensitive to the initial

value. The approach was published in Paper D and it was later used for segmenting the vuggy carbonate rock image data-sets.

Part I

Frequency domain filtering of fluid flow images

Chapter 1

Introduction

At the *Centre for Integrated Petroleum Research* (CIPR) experimental studies have been performed to visualize and analyze fluid flow properties in porous media. The studies are designed to obtained improved knowledge of oil mobilization processes. The larger overall goal is enhanced oil recovery from real oil reservoirs. The experiments are performed by 2D imaging of the displacement process taking place after injecting some liquid into rock slabs. Also of interest is the flow of gas in porous media and some smaller experiments have been conducted.

The imaging is performed using a custom-built in-house X-ray scanner specifically designed for imaging fluid flow displacement processes in porous media. Often the image quality is poor and the image data is corrupted from severe scanner introduced image artifacts that can vary depending on the current scanner calibration. The introduced artifacts are both periodic and non-periodic vertical and horizontal line artifacts that remains constant within an image time-series. These artifacts makes the data more difficult to analyze as they can both distort and partly occlude the object of interest; the fluid flow.

In this part of the thesis the objective is to improve the presentation of the laboratory X-ray image data-sets. The goal is to attenuate the artifacts without distorting the fluid flow. This can be achieved through advanced image filtering. From image processing theory it is known that due to the repetitive behavior of the artifacts they are likely to be visible in the frequency domain as either frequency peaks or ridges. It is therefore natural to consider frequency domain image filtering. Close evaluation of the frequency domain representation of the images reveals that they are indeed visible as frequency ridges and intricate filters are constructed. The developed procedure yields good results when applied to various image data-sets and the image artifacts are largely removed. This is achieved without distorting the fluid flow, which become more easily recognizable.

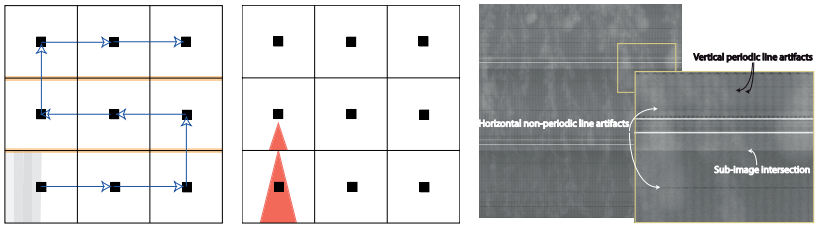
1.1 X-ray imaging of laboratory experiments

Most experiments considered in this work have been performed on rock slabs of size $30 \times 30 \times 2$ cm, although the scanner can handle slabs that are up to 1×1 m. Prior to running the experiments the rock slabs are sealed with an epoxy coating, and fluid inlets and outlets are attached to opposite sides. In the X-ray scanner the slabs are positioned either vertically or horizontally depending on the experiments. To image these slabs the scanner uses sub-section scanning. For rock slabs of size $30 \times 30 \times 2$ cm the scanner require 3×3 sub-sections and it uses about 4-5 minutes to scan the entire region. For the maximum rock size 9×9 sub-sections are needed and the scanning time increase accordingly. The scans are not necessarily taken at regular intervals throughout the experiments. More often they are taken irregularly and the size of the intervals depend on the current displacement process. When there are large changes in the fluid flow scans are taken more often to capture more details. The experimental setup is further described in [29, 28, 30].

Each sub-section is scanned subsequently at different time-points, as is illustrated in Figure 1.1(a). The full 2D image is later generated from the resulting sub-images. Within each sub-section the X-ray source is centered, illustrated by the black squares, and the scanner scans in an overlapping manner from one side to the other, which is illustrated by the vertical gray stripes. It takes about 30 seconds to scan each sub-section, and shifting horizontally from one block to the next is faster than the vertical shift. In the horizontal direction the sub-images are concatenated in a non-overlapping manner. In the vertical direction the sub-images are, however, concatenated in an overlapping manner as they partly covers the same region. This is illustrated by the two orange lines. Ideally image registration should have been used to automatically discard overlapping regions. This is, however, beyond the scope of this work. Instead a trained user has manually set the cutoffs to register the sub-images. This leads to a slightly blocky appearance at the intersections in the vertical direction. The scanning procedure is repeated in time and yields a 2D image time-series.

Since sub-section scanning is applied the fluid flow might move faster than the scanner. This can lead to an inaccuracy in the imaged flow at the intersections, as is illustrated in Figure 1.1(b). In particularly, this is a problem when imaging fingering effects as fingers have a tendency to move rapidly through the domain. It is desirable to correct for the time difference within each 2D image by shifting the sub-images in time. Due to large and possibly non-uniform sampling interval and a relatively small set of time-points, obtaining an accurate result might be difficult. Further investigations are left for future work.

In Figure 1.1(c) a typical water flooding scan, where part of the image has been magnified for better viewing, is shown. Clearly visible are the typical peri-

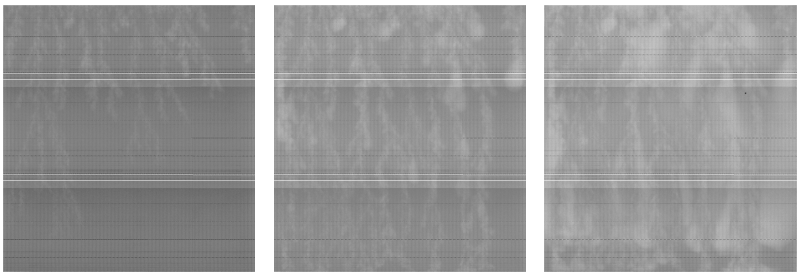


(a) (b) (c)

Figure 1.1: In (a), the X-ray image acquisition procedure is illustrated. A 2D image is generated by sub-section scanning, where each section is scanned at different time-points. Within each section the X-ray source is centered indicated by the black squares. Each section is scanned in an overlapping manner from one side to the other, illustrated by the gray lines. This cause periodic vertical line artifacts that are visible in the resulting images. Vertically the sub-sections are scanned in a slightly overlapping manner, illustrated by the two orange lines. The sub-images are concatenated in a post-processing step. In (b), it is illustrated how fingers can move faster through the domain than the scanner, which might cause an inaccuracy in the imaged fluid flow. In (c), a sample image is shown, in which part of the image has been magnified. The periodic vertical line artifacts are due to the scanning procedure, while the horizontal non-periodic artifacts depends on the current scanner calibration. Two dark horizontal lines are also visible. These are due to sub-image concatination.

odic vertical image artifacts generated by the scanning procedure, as well as the horizontal non-periodic line artifacts that depends on the current scanner calibration. The intersection between the sub-images are visible as two horizontal lines and there is also a visible intensity shift. These artifacts corrupt the image data and can make it difficult to perceive the imaged fluid flow. To bring out the true details attenuation of these artifacts are of importance. Fortunately the artifacts remain constant throughout an image time-series. Therefore filters can be constructed for a single image and then applied to the remaining images.

The scanner generates two types of image file formats, JPEG and TIFF images. The JPEG image data-sets are thresholded by the scanner to reduce the influence of outliers prior to compression. They are given with unsigned 8-bit integer precision, which gives them an intensity range of [0, 255]. In this process the intensity range is largely reduced, which results in poorer intensity resolution. The TIFF images are uncompressed and given with unsigned 16-bit integer precision, thus they have intensity range $[0,\ 65\,535]$. Most intensity values corresponding to the depicted fluid flow are, however, located in the lower part of the intensity range. There are only a few outliers from image artifacts that are scattered in the remaining part. Therefore, the intensity range can be reduced without reducing the



(a) (b) (c)

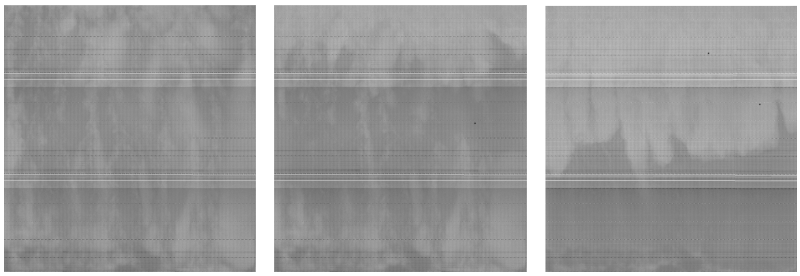
Figure 1.2: Selected images from a water flooding experiment. Observe in (a) that fingering behavior starts early in the experiments. During the experiment some of these fingers are reinforced and water channels are formed, as can observed in (b) and (c).

accuracy. In this work both JPEG and TIFF images are filtered. Similar results are obtained for both filetypes.

1.2 Fluid flow experiments

The image data considered in this work are mostly from experiments conducted on fairly homogenous Bentheimer sandstone acquired from an outcrop in Germany. The experiments were constructed to obtain improved insight into fluid flow properties, in particular how heavy-oil is mobilized by water and polymer flooding. Prior to conducting the experiments, if not stated otherwise, the sandstones were saturated with approximately 90% oil and 10% water to mimic the characteristics of oil reservoirs. Aging is performed to mimic reservoir wettability conditions. The experiments typically starts by oil mobilization using water flooding, which is shown in Figure 1.2. There it is seen how thin sharp fingers develops early on in the experiment and continues to grow throughout the experiment. Fingers occur since water has lower viscosity than oil. Other parameters influencing fingering development is capillary forces and rock heterogeneity. From these fingers other fingers are created mostly by splitting at the tip, and the fingers grow both in width and length. During the experiment some of these fingers are reinforced at the expense of other fingers and water channels are created. The fingers continue to spread throughout the experiments. Nevertheless, the oil recovery from areas originally bypassed by the fingers is poor, causing injected fluid to being mostly produced through the water channels [29, 28, 30]. This was also found in [36] where fluid flow experiments were performed on sandpacks.

After water flooding polymer flooding is often performed. Polymer is a term



(a) (b) (c)

Figure 1.3: Selected images from a polymer flooding experiment. In (a), initially the water channels from a water flooding experiment are clearly visible. Then in (b), observe that a polymer bank has formed, indicating that polymer mobilizes oil bypassed by the water flooding. The polymer moves slightly faster through the water channels. In (c), it is observed that the polymer continues to move slightly faster through the water channels while maintaining the polymer bank.

used about water that is made more viscous by adding polymer molecules. The polymer flooding yields, due to higher viscosity, improved sweep of the domain as can be observed in Figure 1.3. It is clearly able to access regions that were originally bypassed by the water flooding and a polymer bank quickly develops. The oil is being mobilized through the water channels and oil recovery is improved. In fact, the oil recovery will in most cases nearly double compared to water flooding. The fingering effect is reduced since it depends on the viscosity ratio between the mobilized oil and polymer solution. Thus, the amount of fingering will depend on the experiment. There will be less fingering when the polymer solution and displacing oil has low viscosity ratio, for higher viscosity ratio fingering behavior will increase. When the viscosity ratio is low more oil is recovered and a better sweep of the domain is observed [30]. Also in [36] it was found that during polymer flooding the oil is being mobilized into the water channels by choosing the path of least resistance. A better sweep of the domain was obtained and the oil recovery nearly doubles when compared to the water flooding.

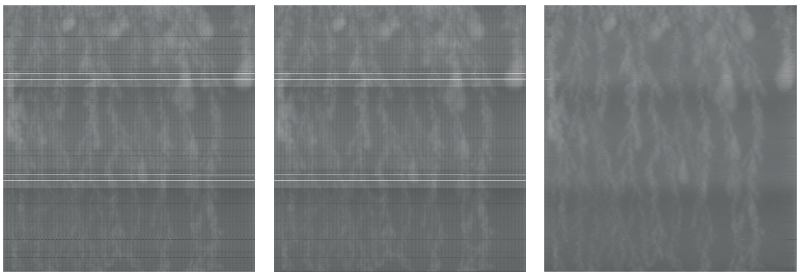
Chapter 2

Filtering of laboratory X-ray images

The X-ray image data-sets considered in this work are corrupted by severe scanner introduced a periodic horizontal and periodic vertical line artifacts. The filtering procedure need therefore attenuate both periodic and a periodic artifacts without distorting the fluid flow. Due to the periodic character of parts of the artifacts it is natural to consider frequency domain image filtering. These artifacts are difficult to attenuate using filtering techniques in the image domain.

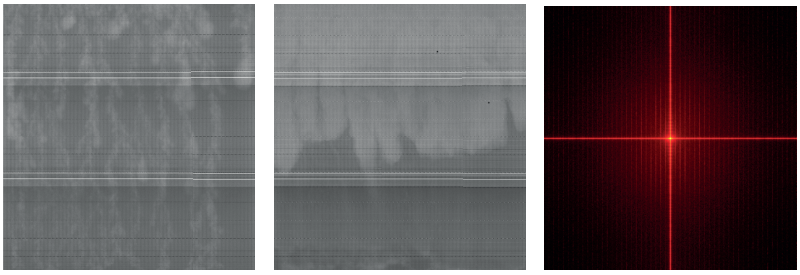
In Figure 2.1 the performance of a standard image filtering technique versus frequency domain filtering is evaluated. There, in (a), a typical input image from a water flooding experiment is shown in which the characteristic line artifacts are clearly visible. In (b), the image has been filtered using a median filter. It is a standard windowing approach that replaces the center pixel by the median within the window [11]. Visually median filtering does not appear to remove any artifacts, although close examinations reveals that some of the artifacts have indeed been removed. In (c), the result of frequency domain filtering in which the artifacts frequency components have been attenuated using the procedure developed in this work is shown. There, the image artifacts are largely removed.

In Figure 2.2 two typical fluid flow images are displayed. In (a), the image is from a water flooding experiment and in (b), it is from a polymer flooding experiment. The characteristic horizontal and vertical line artifacts are visible in both images. In (c), a typical frequency domain representation of the images is shown. The image artifacts are reflected in certain frequency components, which implies that they can be attenuated in the frequency domain by precisely constructed filters. The periodic vertical image artifacts are reflected in the horizontal midline ridge. Parts of the vertical fluid flow will also be represented by this ridge. The horizontal aperiodic artifacts is more complicated. Fortunately they are visible as the multiple nearly periodic vertical frequency ridges seen throughout the fre-



(a)

Figure 2.1: In (a), a typical input image is shown, which clearly has severe vertical and horizontal line artifacts. In (b), the image has been filtered using a median filter. Visually, there is little difference from the original image. In (c), frequency domain image filtering has been applied to attenuate the artifacts. There the artifacts are largely attenuated.



(a) (b) (c)

Figure 2.2: In (a) and (b) typical input images from water and polymer experiments are displayed, respectively. The characteristic line artifacts are clearly visible. In (c), a typical frequency domain representation of the images is shown. The single horizontal frequency ridge corresponds in part to the periodic vertical image artifacts. Parts of the vertical fluid flow is also represented by this ridge. The multiple vertical frequency ridges corresponds to the aperiodic horizontal image artifacts.

quency domain. To filter the image data these ridges need first be detected and then attenuated by precisely constructed filters. The challenge lies in locating the ridges accurately and also in constructing precise filters that do not interfere with the the fluid flow. The developed procedure has proven to produce good results and much of the line artifacts are removed, bringing out the true details in the image data.

In the following, an introduction to frequency domain filtering is given prior

to presenting the constructed frequency domain filters. The following exposition is largely based on the two textbooks [11, 24].

2.1 Freqeuncy domain image filtering

In the image domain an image $f(x,\, y)$ is often filtered by convolving $f(x,\, y)$ by a folded sliding filter $h(x,\, y)$. The *linear convolution sum* is given by

$$g(x,\, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,\, n) h(x - m,\, y - n)\,, \tag{2.1}$$

which is a sum over all the pixels in $f(x,\, y)$, and M and N is the number of bins in the x- and y-direction, respectively. The filter $h(x,\, y)$ is of size $P \times Q$ and it is usually much smaller than $f(x,\, y)$. Depending on the size of the filter, the size of the convolution output $g(x,\, y)$ will be larger than $f(x,\, y)$. In fact $g(x,\, y)$ will have size $A \times B$, where A and B equals

$$A = M + P - 1 \quad \text{and} \quad B = N + Q - 1\,, \tag{2.2}$$

as is illustrated for one-dimensional (1D) and/or 2D sequences in [11, 24].

Convolution is a time consuming operation, especially for large data-sets. Fortunately *circular convolution* in the image domain is equivalent to multiplication in the frequency domain [24]. Thus we have

$$f(x,\, y) \star h(x,\, y) \Leftrightarrow F(u,\, v) H(u,\, v)\,, \tag{2.3}$$

where $\star$ indicate convolution. Frequency domain filtering will often be faster than the corresponding image domain approach. The *circular convolution sum* is given by

$$g(x,\, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h((x - m)_M, (y - n)_N)\,, \tag{2.4}$$

which is similar to the linear convolution sum in eq. (2.1). It differs only in the formulation of the filter $h(x,\, y)$ in which the indices are computed by taking the modulo of M and N. For circular convolution, both $f(x,\, y)$ and $h(x,\, y)$ need to be of the same size. The 2D circular convolution sum can be derived in a similar manner as the 1D circular convolution sum in [24].

Linear and circular convolution sums are similar, but not identical. To ensure that circular convolution yields the same result as linear convolution *zero padding*

is performed on both $f(x,\, y)$ and $h(x,\, y)$ so that the number of bins in each direction fulfill eq. (2.2). Adding additional zeroes will not influence the output, so the formal requirement is that the number of bins in each direction fulfill

$$A \geq M + P - 1 \quad \text{and} \quad B \geq N + Q - 1\,. \tag{2.5}$$

The zeroes are appended to the end of the image in both the x- and y-direction. Zero padding will also ensures better display in the frequency domain due to more bins. It does not, however, provide any additional information about the spectrum [24]. *Wraparound errors* occur when not enough zeroes have been appended to $f(x,\, y)$ and $h(x,\, y)$ [11]. They are a result of pixels from one side of the image influencing the convolution output at the opposite side. Circular convolution is now no longer equivalent to linear convolution. Thus, zero padding with a sufficient number of zeros is important both to avoid errors and to ensure that circular convolutions yields equivalent result when compared to linear convolution [24].

To filter an image in the frequency domain, the frequency domain representations of both the image and the filter are needed. The frequency domain representation of and image $f(x,\, y)$ is computed by the *discrete Fourier transform* (DFT),

$$F(u,\, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,\, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}\,. \tag{2.6}$$

The same approach can be used to compute the frequency domain representation $H(u,\, v)$ of a filter $h(x,\, y)$. A similar transform is applied to transfer the image back into the image domain, using the *inverse discrete Fourier transform* (IDFT),

$$f(x,\, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,\, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}\,. \tag{2.7}$$

The two transforms make up the Fourier transform pair [11].

The frequency domain representation $F(u,\, v)$ can also be represented on polar form,

$$F(u,\, v) = |F(u,\, v)| e^{j\phi(u,\, v)}\,, \tag{2.8}$$

since it contains both a real and imaginary part. It is made up of two terms, the *Fourier spectrum* $|F(u,\, v)|$ and *phase angle* $\phi(u,\, v)$. The Fourier spectrum $|F(u,\, v)|$ is computed by

$$|F(u,\, v)| = \sqrt{\mathcal{R}^2(u,\, v) + \mathcal{I}^2(u,\, v)}\,, \tag{2.9}$$

where $\mathcal{R}(u,\, v)$ and $\mathcal{I}(u,\, v)$ are the real and imaginary part of $F(u,\, v)$, respectively. It holds much of the image intensity information and it is analyzed to locate the frequency components of the image artifacts. Filters are usually constructed directly in the frequency domain as precise filters are more easily constructed there than in the image domain. The phase angle $\phi(u,\, v)$ holds much of the information related to object location and it is computed by

$$\phi(u,\, v) = \tan^{-1}\left(\frac{\mathcal{I}(u,\, v)}{\mathcal{R}(u,\, v)}\right). \tag{2.10}$$

It is difficult to analyze and is usually not modified during a filtering operations [11]. In this work filters are always constructed in the frequency domain and only real filters that do not modify the phase angles are considered.

The frequency domain representation $F(u,\, v)$ is an infinitely periodic sequence in both the u- and v-direction. Therefore, it is sufficient to look at a single period and it is customary to work with the period containing the origin. The zero frequency, $F(0,\, 0)$, is commonly the largest frequency component and it is proportional to the average image intensity value since $F(0,\, 0) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,\, y)$. The spectrum is symmetric about the origin and by shifting the data,

$$F(u,\, v) = F(u - M/2,\, v - N/2)\,, \tag{2.11}$$

the origin is ensured to lie in the center of the evaluated frequency spectrum, $F(M/2,\, N/2)$. This facilitate visual evaluation of the spectrum and also filter construction. Prior to computing the IDFT it is shifted back so that the origin once again lies at the zero frequency, $F(0,\, 0)$ [11].

Filtering in the frequency domain is performed by computing the product $F(u,\, v)H(u,\, v)$, which according to eq. (2.3) is equivalent to circular convolution in the image domain. This implies that

$$g(x,\, y) = \mathcal{IDFT}\{F(u,\, v)H(u,\, v)\} \tag{2.12}$$

should be equivalent to the output of the circular convolution sum from eq. (2.4). In this work the filter $H(u,\, v)$ is, however, always constructed directly in the frequency domain. Therefore, even if the input image $f(x,\, y)$ has been zero padded prior to computing $F(u,\, v)$ some wraparound errors occur since $H(u,\, v)$ has not been zero padded. To limit their effect the input image can be extended in all directions, prior to zero padding, by replicating the boundary pixels by a given number of pixels [11].

2.2 Filter construction

The filter $H(u,\, v)$ is constructed directly in the frequency domain by combining different types of filters. Successful filtering depends on accurate attenuation of frequencies originating from image artifacts. In this work three different filters are combined to attenuate the image artifacts; a *Gaussian lowpass filter*, *vertical notch reject filters* and *horizontal notch reject filters*. In the following, the filter construction is presented.

2.2.1 Common filter parameters

The three different types of constructed filters use two common filter parameters, the *cutoff frequency* D_0 and the *distance map* $D(u,\, v)$. From signal processing theory it is known that the Fourier spectrum is periodic and it is unique over the interval $(-\pi,\, \pi)$ (or $(0,\, 2\pi)$) [24]. Therefore, in this work, both D_0 and $D(u,\, v)$ lies within range $[0,\, \pi]$. The cutoff frequency D_0 is given on the form of

$$D_0 = \pi \frac{1}{f_0}\,, \quad \text{for } f_0 \geq 1\,, \tag{2.13}$$

which ensures that $D_0 \in [0, \pi]$. The distance map $D(u,\, v)$ is computed by

$$D(u,\, v) = \left((u - M/2)^2 + (v - N/2)^2 \right)^{\frac{1}{2}}, \tag{2.14}$$

where u and v are the frequency domain coordinates. The value of $D(u,\, v)$ is the distance from index $(u,\, v)$ to the center of the domain, and M and N is the number of bins in the u and v direction, respectively [11]. The map $D(u,\, v)$ is scaled into range $[0,\, \pi]$.

2.2.2 Gaussian lowpass filtering

The goal of lowpass filtering is to blur an image by attenuating high-frequencies while passing low frequencies. This preserves low frequencies related to slowly varying image intensity component, while blurring the components related to high frequencies such as object boundaries and image artifacts. To preserve image object information lowpass filtering should not be too severe. In this work a Gaussian lowpass filter is applied. It does not introduce any ringing artifacts and yields a smooth filter without any side-lobes in both the image and frequency domain. The Gaussian lowpass filter is given by

$$H_{Lp}(u,\, v) = e^{-\frac{D^2(u,v)}{2D_0^2}}\,. \tag{2.15}$$

Thus, the cutoff frequency D_0 is used as the standard deviation controlling the width of the lobe and therefore also the amount of smoothing [11].

2.2.3 Notch filtering

Notch filters are used for stopping selected frequencies, while passing the remaining ones, or vice versa. They are created with narrow reject bands (or passbands) and are symmetric about the origin. In general, notch filters are given on the form of

$$H_L(u,\, v) = \prod_{i=1}^{L} H_i(u,\, v) H_{-i}(u,\, v)\,, \tag{2.16}$$

where L are the number of notch filter pairs. The two filters $H_i(u,\, v)$ and $H_{-i}(u,\, v)$ are centered with respect to the origin at $(u_i,\, v_i)$ and $(-u_i,\, -v_i)$, respectively. Their distance maps are therefore given by

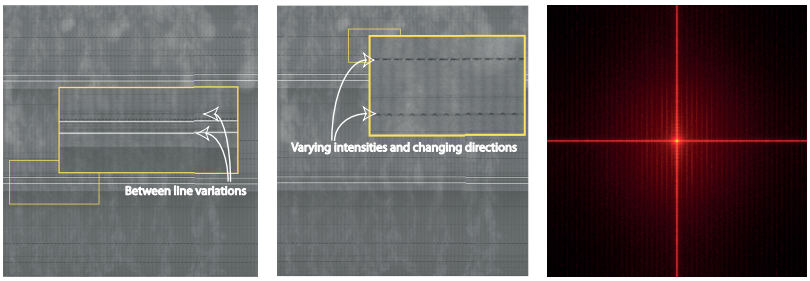
$$D_i(u,\, v) = \left((u - M/2 - u_i)^2 + (v - N/2 - v_i)^2\right)^{\frac{1}{2}} \tag{2.17}$$

$$D_{-i}(u,\, v) = \left((u - M/2 + u_i)^2 + (v - N/2 + v_i)^2\right)^{\frac{1}{2}}. \tag{2.18}$$

The point $(M/2,\, N/2)$ corresponds to the origin of the frequency domain, and u_i and v_i the displacement in the u and v direction, respectively [11]. Both maps are scaled into range $[0,\, \pi]$.

2.2.4 Vertical notch filtering

Vertical notch reject filters are applied to attenuate frequencies coming from the aperiodic and non-homogeneous *horizontal* line artifacts. The intensities of the artifacts vary both between lines and within a single horizontal line. Close inspection of individual horizontal lines reveals within line features with changing directions, as can be seen in Figure 2.3(a)-(b). This makes it difficult to determine their exact frequencies. In the frequency spectrums there are multiple *vertical* frequency ridges that run through the entire spectrum, as is observed in (c). These frequency ridges indicate that there are frequency components corresponding to artifacts in those locations. Due to the within line directional changes they correspond to the horizontal artifacts. The fluid displacement changes slowly and correspond to mostly low frequency components close to the spectrums origin. The origin also holds much of the image intensity information [11] and care must be taken not to modify it during filtering. Therefore the attention is focused on attenuation of the vertical ridges while leaving the origin unchanged.



(a) (b) (c)

Figure 2.3: Horizontal line artifacts. In (a) and (b) two different sections of an image has been magnified. Observe that the horizontal artifacts are both aperiodic and non-homogenous. The intensities varies both within and between the horizontal lines, and various within line features can be seen. In (c), the images frequency spectrum is shown. The multiple vertical ridges corresponds to the horizontal line artifacts. The frequencies of the vertical periodic image artifacts lies along the horizontal midline ridge.

To attenuate the vertical frequency ridges, 1D Gaussian reject filters are computed for each ridge and for each row, symmetrically about the spectrums origin. The filters are constructed in a pair-wise manner by

$$H_i(u,\, v) \;=\; 1 - e^{-\frac{D_i^2(u)}{2D_0^2}}, \quad D_i(u) \in [0,\, \pi],\, 0 < D_0 \leq \pi\,, \tag{2.19}$$

$$H_{-i}(u,\, v) \;=\; 1 - e^{-\frac{D_{-i}^2(u)}{2D_0^2}}, \quad D_{-i}(u) \in [0,\, \pi],\, 0 < D_0 \leq \pi\,, \tag{2.20}$$

for a common D_0. The computations are repeated for all rows, $v = 0, \ldots, N-1$. The two 1D distance maps are independent of row v and are given by

$$D_i(u) \;=\; u - M/2 - u_i\,, \tag{2.21}$$

$$D_{-i}(u) \;=\; u - M/2 + u_i\,. \tag{2.22}$$

For both maps u_i is the displacement of the origin so that $D_i(u) = 0$ and $D_{-i}(u) = 0$ lies at the center of a ridge. The center of each ridge is semi-automatically detected using the procedure outlined in Appendix A. Both maps are scaled into range $[0,\, \pi]$ as before. Thus, the following vertical notch reject filter is obtained,

$$H_{Vn}^*(u,\, v) = \prod_{i=1}^{L} H_i(u,\, v) H_{-i}(u,\, v)\,, \tag{2.23}$$

where L is the number of notch filter pairs. An example of this filter can be observed in Figure 2.4(a). It does not include midline filtering and it is used for

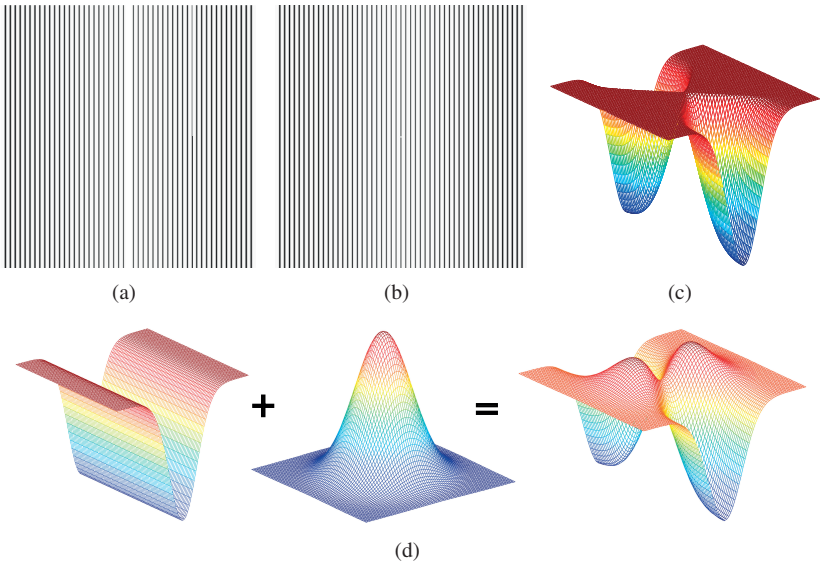


Figure 2.4: Vertical notch filtering. In (a), a vertical notch filter that do not include midline ridge filtering is shown. It is used for filtering image data-sets with horizontal fluid flow. In (b), the same filter is shown only now including midline ridge filtering. It is used for filtering data-sets with vertical fluid flow. In (c), a close up view of the midline filters center point is shown. It attenuates the vertical midline while leaving the origin unchanged. In (d), the construction of the midline reject filter is illustrated. To obtain the filter in (c) all values greater than one are set to one.

filtering data-sets with horizontal fluid flow since midline filtering would distort the flow.

For data-sets with vertical fluid flow, vertical midline filtering does not distort the flow. A filter is therefore created that attenuates the midline frequency ridge while leaving the spectrums origin unchanged. Initially the idea was to gradually increase the constant f_0 in eq. (2.13) to create a gradually narrowing midline reject filter as the filter approaches the origin. This was the approach used in Paper A. Even though this approach left the origin unchanged and good results were obtained, a low-frequent sinusoid was introduced by the filter. It was therefore abandoned. Instead the midline notch filter was created by combining the midline reject filter,

$$H_0(u,\, v) = 1 - e^{-\frac{D_0^2(u)}{2D_0^2}}, \quad D_0(u) \in [0,\, \pi],\, 0 < D_0 \leq \pi\,, \tag{2.24}$$

where $D_0(u)$ is given by eq. (2.21) for $u_0 = 0$, with a 2D Gaussian lowpass filter $H_{Lp}(u,\, v)$, which yields

$$H'(u,\, v) = H_0(u,\, v) + H_{Lp}(u,\, v)\,, \quad \text{for common } D_0\,. \tag{2.25}$$

This process is illustrated in Figure 2.4(d). It is observed that adding the two filters yields a filter that leaves the origin unchanged and attenuates the vertical midline, but also include two symmetric peaks that have intensity values above one. The constructed filters should, however, have intensity values in the range [0 1] to attenuate certain frequencies while leaving the remaining frequencies unchanged. To preserve this property all intensities above one are set to one,

$$H'(u,\, v) = \begin{cases} 1, & \text{if } H'(u,\, v) > 1 \\ H'(u,\, v), & \text{otherwise} \end{cases}\,. \tag{2.26}$$

In (c), a close up view of the constructed midline filter center point is shown. Although this approach introduces a slight discontinuity in the filter, the results does not have any visible introduced artifacts. It attenuate the image artifacts while leaving other image features unchanged. The combined vertical notch filter, shown in Figure 2.4(b), is given by

$$H^{**}_{Vn}(u,\, v) = H^{*}_{Vn}(u,\, v) \cdot H'(u,\, v)\,. \tag{2.27}$$

It is used for filtering data-sets with vertical fluid flow.

2.2.5 Horizontal notch filtering

Horizontal notch reject filters are applied to attenuate frequencies coming from the periodic *vertical* image artifacts. These artifacts lies along the horizontal midline frequency ridge, as can be observed in Figure 2.3(c). This ridge is not monotone, but have frequency peaks at the intersection with the vertical frequency ridges, which are more visible closer to the origin.

For data-sets with horizontal fluid flow a horizontal *midline* reject filter can be applied since the fluid flows frequency components lies along the vertical midline frequency ridge. This midline reject filter can be constructed in a similar manner as the vertical midline reject filter. It is created by combining a midline reject filter with a 2D Gaussian lowpass filter $H_{Lp}(u,\, v)$. The midline reject filter is given by,

$$H_0(u,\, v) = 1 - e^{-\frac{D^2(v)}{2D_0^2}}\,, \quad D(v) \in [0,\, \pi],\, 0 < D_0(u) \leq \pi\,, \tag{2.28}$$

which is similar to the filter in eq. (2.24). It is now constructed in a column-vise manner and the distance map $D(v)$ depends on v, computed by

$$D(v) = v - N/2\,. \tag{2.29}$$

(a) (b)

Figure 2.5: Horizontal notch filtering. In (a), a horizontal midline filter is shown. It is used in the case of horizontal fluid flow. In (b), multiple combined 2D Gaussian reject filters is shown, situated at the intersections between the spectrums vertical and horizontal frequency ridges. It is used in the case of vertical fluid flow.

It is scaled into range $[0,\, \pi]$. The combined filter is then computed by

$$H'(u,\, v) = H_0(u,\, v) + H_{Lp}(u,\, v)\,, \quad \text{for common } D_0\,. \tag{2.30}$$

This yields a filter with two frequency peaks that have frequency values greater than one, as was the case for vertical midline filter. To ensure that certain frequencies are not reinforced all values greater than one is set to one,

$$H^*_{Hn}(u,\, v) = \begin{cases} 1, & \text{if } H'(u,\, v) > 1 \\ H'(u,\, v), & \text{otherwise} \end{cases}\,. \tag{2.31}$$

The resulting filter is displayed in Figure 2.5(a), and it attenuates the horizontal midline frequency ridge while leaving the origin unchanged.

For data-sets with vertical fluid flow the horizontal midline filter will modify the fluid flow. A second type of filter along the horizontal midline is therefore constructed. Since the horizontal midline ridge have frequency peaks at the intersection with the vertical frequency ridges, multiple 2D Gaussian reject filters are constructed and positioned at the intersections. The horizontal midline notch filters are now given by

$$H^{**}_{Hn}(u,\, v) = \prod_{i=1}^{L} \left(1 - e^{-\frac{D_i^2(u,v)}{2D_0^2}}\right)\left(1 - e^{-\frac{D_{-i}^2(u,v)}{2D_0^2}}\right), \tag{2.32}$$

where L is the number of notch filter pairs, D_0 is the cutoff frequency, and $D_i(u,\, v)$ and $D_{-i}(u,\, v)$ are the distance maps given by eq. (2.17) and eq. (2.18), respectively. The center point of each reject filter is found by the procedure outlined in Appendix A. The resulting filter is shown in Figure 2.5(b), and it attenuates the vertical image artifacts while leaving the vertical fluid flow unchanged. Observe that the filters kernel size increases away from the origin. This is due to the influence of the neighboring reject filters. It does not appear to influence the results in any noticeable way.

2.2.6 Constructed combined filter

The constructed combined filter, $H(u,\, v)$, used in the filtering procedure varies depending on the data-set. For all data-sets a Gaussian lowpass filter $H_{Lp}(u,\, v)$ from eq. (2.15) is included. The notch filters does, however, vary depending on the direction of the flow. For each time-series the filter need only be computed once for a single image. This filter can be applied to the remaining images as the line artifacts remain constant throughout an experiment.

In the case of *vertical fluid flow*, the flows frequency components are located along the horizontal midline. To attenuate the vertical line artifacts while preserving the fluid flow, multiple Gaussian reject filters are included along the midline that are given by $H^{**}_{Hn}(u,\, v)$ from eq. (2.32). Vertical notch filtering does not influence the fluid flow, therefore the vertical notch filter $H^{**}_{Vn}(u,\, v)$ from eq. (2.27) is used. The resulting applied filter is therefore

$$H(u,\, v) = H_{Lp}(u,\, v) \cdot H^{**}_{Hn}(u,\, v) \cdot H^{**}_{Vn}(u,\, v)\,. \tag{2.33}$$

The combined filter $H(u,\, v)$ used for filtering data with *horizontal fluid flow* is constructed in a similar manner. For these data-sets the fluid flows frequency components are located along the vertical midline. This midline is therefore omitted when constructing $H(u,\, v)$. Thus the included vertical notch filter is $H^{*}_{Vn}(u,\, v)$ from eq. (2.23). Horizontal notch filtering will not influence the fluid flow, therefore the horizontal midline filter $H^{*}_{Hn}(u,\, v)$ from eq. (2.31) is applied. The resulting combined filter is therefore

$$H(u,\, v) = H_{Lp}(u,\, v) \cdot H^{*}_{Hn}(u,\, v) \cdot H^{*}_{Vn}(u,\, v)\,. \tag{2.34}$$

Chapter 3

Results and conclusion

The developed procedure is applied for filtering X-ray image data-sets with either vertical or horizontal fluid flow. The main focus has been on filtering image data-sets from experiments performed on Bentheimer sandstones. The procedure works for data-sets from other types of experiments as well, and the result of filtering an image from a Berea sandstone experiment is included. Good results are obtained and the procedure removes much of the vertical and horizontal line artifacts. This is true for both for TIFF and JPEG images. Here TIFF images are used due to better intensity resolution. In Paper A JPEG images were used.

3.1 Filtering results

In Figure 3.1 the result of filtering a water flooding experiment on a Bentheimer sandstone is displayed. On the *upper row* selected input images are shown. On the *middle row* the corresponding output after filtering with filter $H(u,\, v)$. Since the flow runs vertically the combined filter from eq. (2.33) is used. On the *lower row* the difference between input and output images are shown. Observe that good results are obtained. The filtering procedure attenuates the vertical and horizontal line artifacts while preserving the fluid flow. This is confirmed by the difference plots. The horizontal intensity shifts visible in the input images are also present in the results. Wraparound errors are not completely eliminated since $H(u,\, v)$ is constructed in the frequency domain. They can be observed along the vertical image boundaries where the horizontal line artifacts are more visible than to the interior. These errors are handled by image extension, which is covered further in Section 3.2.

Image filtering of other time-series yields similar results as in the previous example. In Figure 3.2 the result of filtering a polymer flooding experiment on a Bentheimer sandstone is shown. The filter $H(u,\, v)$ is constructed by eq. (2.33)

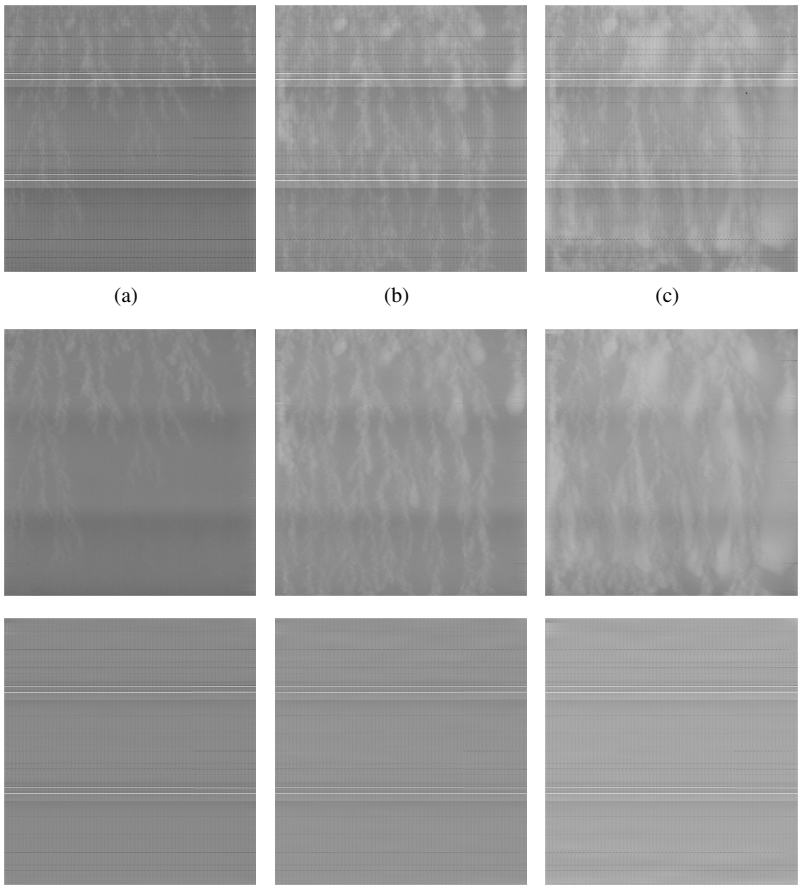


Figure 3.1: Water flooding experiment. The *upper row* displays selected input images, the *middle row* the corresponding output after frequency domain filtering and the *lower row* the difference between input and output images. Observe that line artifacts are largely attenuated while preserving the fluid flow, which is confirmed by the difference plots.

as in the previous example, and also here the filtering procedure removes much of the vertical and horizontal line artifacts while preserving the fluid flow. As for the water flooding experiment, filtering yields good results and artifacts are largely attenuated. Also here some wraparound errors are visible along the vertical image boundaries. The horizontal intensity shifts seen in the input images are also visible in the output images.

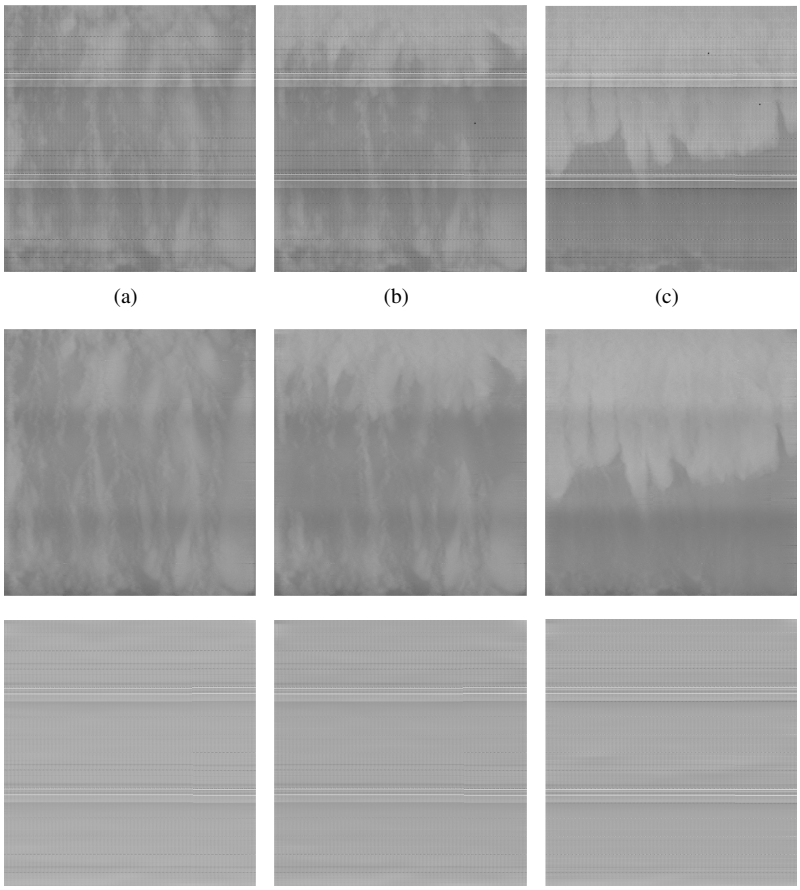
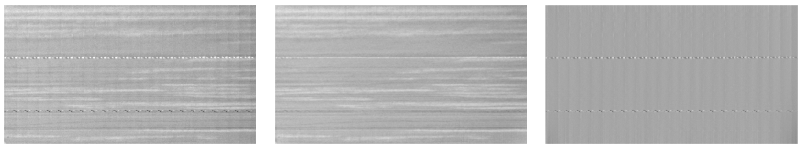


Figure 3.2: Polymer flooding experiment. The *upper row* displays selected input images, the *middle row* the corresponding output after frequency domain filtering and the *lower row* the difference between input and output images. Observe that the line artifacts are attenuated while preserving the fluid flow.

In Figure 3.3 the performance of the method when applied to an image data-set acquired from a gas flooding experiment on a heterogeneous Berea sandstone is shown. For this experiment the rock slab was completely saturated by water prior to flooding. In (a), a selected slice of the input image data-set is shown. It is observed that gas flows horizontally, choosing the way of least resistance. The typical horizontal and vertical image artifacts are also visible. Since the flow



(a) (b) (c)

Figure 3.3: Gas flooding experiment. In (a), an input image is shown and in (b), the resulting output after filtering. Observe that the horizontal and vertical line artifacts have been largely attenuated while preserving the fluid flow. In (c), the difference image confirms that the filter attenuate artifacts while preserving the fluid flow.

runs horizontally the combined filter $H(u, v)$ from eq. (2.34) is applied. In (b), the resulting output after frequency domain filtering is shown. It is observed that the filtering procedure successfully attenuates most of the vertical and horizontal image artifacts. Compared to the two previous examples the horizontal artifacts are not as well attenuated since the flow runs horizontally. Good results are, however, obtained and there are few wraparound errors. In (c), the difference image confirms that the image artifacts are attenuated while preserving the fluid flow.

3.2 Wraparound error handling

The influence of wraparound errors are made less severe by image extension in which the input image boundary pixels are replicated a certain number of times prior to zero-padding. In Figure 3.4 the effect of image extension is illustrated. In (a), the input image is shown, and in (b)-(f), the filtered outputs after extending the input image by 0, 100, 200, 400 and 600 pixels, respectively, in all directions. Wraparound errors are observed along the horizontal and vertical image boundaries. In (b), both the vertical and horizontal line artifacts are clearly visible along the image boundaries. Along the horizontal image boundaries a pixel intensity drop is observed. It is due to vertical midline filtering and is also visible in (c). The influence of wraparound errors lessen as the input image is extended by more pixels and become difficult to perceive. Visually there is little difference in the outputs when the input image has been extended by 200 pixels or more. The interior of the images does not appear to be influenced by image extension and there is little difference between the results.

In Figure 3.5 vertical and horizontal cross section plots of the input and the filtered results confirms this trend. Observe that filtering greatly reduce the noise level in the image signal compared to the input. Image filtering without first performing image extension results in an intensity drop at the image boundaries,

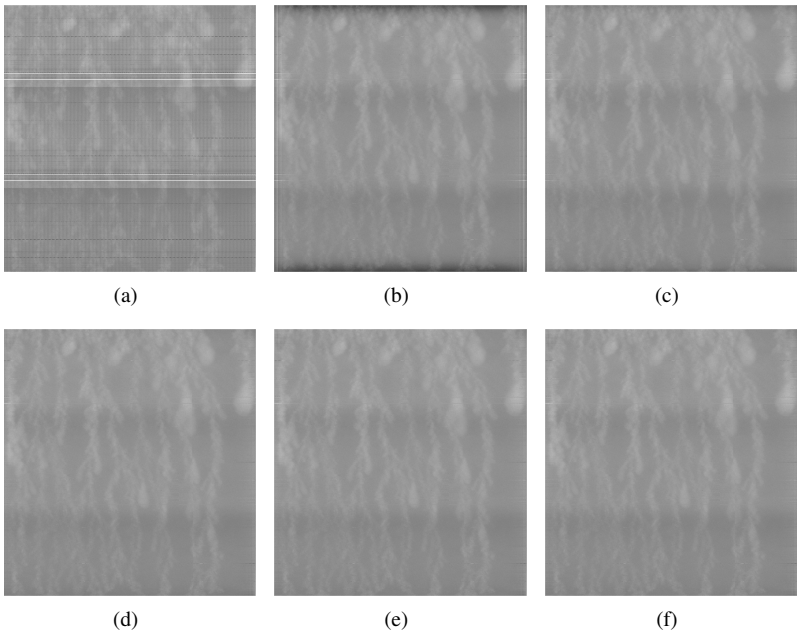
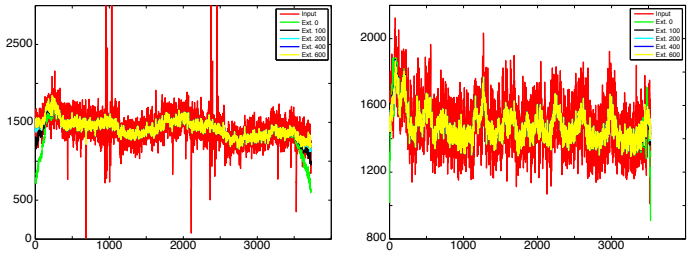


Figure 3.4: Wraparound error handling. In (a), the input image is shown, and in (b)-(f), the filtering output after extending the image by 0, 100, 200, 400 and 600 pixels, respectively, in all directions. Observe that close to the image boundaries the horizontal and vertical line artifacts are better attenuated when the input image has been extended by more pixels. To the interior of the images few differences are observed. At the horizontal image boundaries a drop in pixel intensities are observed in (b)-(c). This is due to vertical midline filtering. Visually, there is little difference when the input image is extended by 200 pixels or more.

which is clearly visible in both cross section plots. This intensity drop is reduced, or eliminated, as the input image is extended by more pixels. For the vertical cross section plots in (a) this intensity drop is also visible in the outputs when the input image has been extended by up to 200 pixels. When the input image has been extended by 400 pixels or more, the cross section plots are similar. For the horizontal cross section view in (b) the outputs are similar as long as they have been extended by more than 100 pixels. To the interior of the image there is little difference between cross section plots of the output images. This indicates that the filtering procedure works equally well to the interior regardless of image extension.



(a) (b)

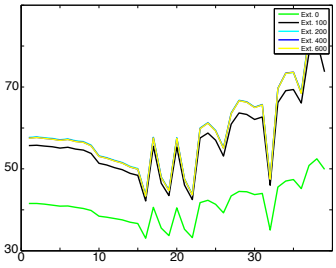
Figure 3.5: Cross section plots. In (a) and (b) vertical and horizontal cross section plots are presented, respectively. The input images pixel intensities have more variation than the pixels coming from the filtered images. When the input image is processed without including extension, the output image experience an intensity drop at the boundaries. Vertically this can also be observed up to extension by 200 pixels.

The *signal-to-noise ratio* (SNR) can be applied to measure how close the filtered image $g(x,\, y)$ is to the input image $f(x,\, y)$ by computing

$$SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{n-1} g(x,\, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{n-1} [g(x,\, y) - f(x,\, y)]^2}\,. \tag{3.1}$$

The closer $g(x,\, y)$ is to $f(x,\, y)$ the larger the SNR will be due to smaller error [11]. Computing SNR for the various extensions is used to evaluate how they influence the filtering result. In Figure 3.6 it is observed that SNR increase until the input image has been extended by 200 pixels, it then reaches a steady state and remains more or less fixed when increasing the number of pixels. Computing the average SNR confirms this. It is greatest when extended by 200 pixels, but does not decrease much when extending by more pixels. Thus, image extension reduces the effect of wraparound errors.

To summarize, it has been illustrated that visually there is little difference in the output images when the input image has been extended by 200 pixels or more. This is also true for SNR computations. However, in the vertical cross section plots the graphs appear to reach steady state when extended by 400 pixels or more. Therefore, to avoid the intensity drop at the horizontal image boundaries all input images are extended by 400 pixels in all directions.



(a)

Extension	Average SNR
0	41.1216
100	56.8255
200	59.2587
400	59.1440
600	59.1070

(b)

Figure 3.6: Signal-to-noise ratio. In (a), observe that SNR increase with increasing image extension until reaching steady state. Little variation is seen after the input image has been extended by 200 pixels or more. This is confirmed in (b) where the average SNR over all images in a single time-series is computed.

3.3 Conclusion

The objective of this work was to improve the presentation of X-ray image datasets of fluid displacement processes acquired using an in-house X-ray scanner. It produces images that are corrupted by both periodic vertical and non-periodic horizontal line artifacts. To improve the image quality advanced image filtering is required. Good results are obtained and the image artifacts are largely attenuated.

The line artifacts are visible in the frequency domain as frequency ridges. Frequency domain filtering is therefore a natural option. The frequency ridges are semi-automatic detected before constructing accurate reject filters. It is sufficient to compute the frequency filter $H(u, v)$ for a single image since the artifacts frequency components remain constant within an image time-series. The filtering procedure largely attenuate both the horizontal and vertical line artifacts without distorting the fluid flow.

While constructing the filters care must be taken not to distort the fluid flow. Since the flow runs either horizontally or vertically parts of the flows frequency components will be located at the vertical or horizontal midline ridge, respectively. Therefore, to ensure that the fluid flow is not modified the constructed filters needs to take the direction of the flow into account.

To obtain satisfactory filtering results wraparound errors need to be handled. In this work they are handled by image extension in which boundary pixels are replicated, followed by zero-padding. This will ensures that wraparound errors have little influence on the resulting filter outputs. Since the filter $H(u, v)$ is constructed in the frequency domain wraparound errors will, however, to a certain

degree always be present.

The filtering procedure can be applied to both TIFF and JPEG images. Good and similar results are obtained for both file-types, although TIFF images have better intensity resolution.

3.4 Future work

The image time-series are generated from sub-section scanning in which each section is scanned at different time-points and later concatenated. Currently, this time difference is not corrected for and the sub-image registration is done manually by a trained user. There are therefore room for some improvements. Firstly, the sub-image registration process should be automated using image registration. Secondly, the time difference within each image slice should be corrected for in the future by shifting the sub-images in time prior to concatenating them. There is also a time difference within each sub-image as they are generated by scanning from one side to the other. Correcting for this time difference will be difficult.

The presented filtering procedure largely attenuates the vertical and horizontal image artifacts without distorting the fluid flow. Wraparound errors are, however, visible at the image boundaries. In the future improved handling of wraparound errors should be included to further limit their influence.

In the future, it would be interesting to segment the filtered data-sets, which will make it possible to do various types of computations on the data. It would for instance be interesting to extract the shape of the fluid flow and measure the length and shape of fingers for different types of experiments. This would yield improved knowledge of fluid flow in porous media and facilitate modeling. Prior to segmentation the horizontal intensity shift present in the image data-sets should be corrected for.

Part II

Segmentation of vuggy carbonate rock images

Chapter 4

Introduction

It is commonly understood that carbonate reservoirs holds much of the world's oil and gas reserves. To better understand fluid flow properties in carbonate rocks pore space characterization is of importance. This is commonly done by laboratory experiments where different rock properties are measured such as porosity, permeability, dispersion characteristics, pore size distributions and capillary pressure. Another approach that is gaining more popularity is to image the samples and compute properties directly from the digital image volumes.

In this part of the thesis the objective is to analyze the performance of various image segmentation techniques applied for segmenting CT scans of vuggy carbonate rocks. Image segmentation is applied to obtain a 3D binary representation of the vuggy pore space. It divides an intensity image into regions based on some given criteria, such as homogeneous voxel intensities. This will provide better insight into the pore spaces connectivity and also allows for the computation of various pore space characteristics such as porosity, pore length distribution and volume distribution. The computations are performed in both 2D and 3D, and a comparison of the results is given.

It can be difficult to differentiate between image phase in intensity images, and the image histograms yield overlapping histogram populations. For high resolution images this makes it challenging to classify voxels at image object boundaries. As the image resolution decreases it becomes increasingly difficult to differentiate between the pore space and rock matter in the image data-sets. The computations show that the segmentation methods perform well for high resolution μCT-scans. When the image resolution decrease a larger portion of the image voxels are wrongly classified. The computed porosity is, due to image resolution, underestimated when compared to the measured porosity from laboratory experiments.

Five different segmentation methods are evaluated. The goal is to determine which method results in the best segmentation at varying image resolutions. The

three first methods are global thresholding techniques, including *basic global segmentation*, as well as the *Ridler and Calvard's* (RC) method [25] and *Otsu's* method [23], which are iterative approaches. These methods computes a hard threshold that is applied for separating the pore space from rock matter. They are included as a reference and the two more complex methods should ideally provide an improvement.

The two remaining methods are thresholding by *indicator kriging* (IK) [20] and the *piecewise constant level set* (PCLS) method [15]. IK is a local thresholding method that segments an image into two populations by using a local spatial covariance estimate and IK, followed by a probability estimate. It is encountered in the literature for segmenting image data-sets of vuggy carbonate rocks (see e.g. [1, 16]). To apply the IK method to our data-sets the method has to be slightly modified to account for three-phase image data-sets. The PCLS method is a region based segmentation method that can detect image object boundaries both with and without gradient information. It segments an image by minimizing an energy functional. The method is valid for multiphase segmentation problems and can be applied directly without modifications to our data-sets. In addition, in this work, a 3D graph cut solver for the PCLS method has been developed. The solver is based on the existing 2D graph cut algorithm presented in [2]. This resulted in a fast solver when compared to the traditional *Euler-Lagrange* approach. It is an accurate and stable solver, which is not very sensitive to the initial value.

4.1 CT imaging of vuggy carbonate rocks

Vuggy carbonate rocks are heterogenous porous media that are mainly composed of carbonate minerals. The pore size vary throughout the rock samples. The smaller pores are not visible to the human eye, although some (but not all) are visible under the microscope. The larger pores are, however, visible without magnifying the samples. The pore space can therefore be divided into two categories, the matrix and vuggy porosity. The matrix porosity is the smaller type of porosity and mainly consists of intergranular or intercrystalline porosity. The vuggy porosity is the visible larger porosity, and can be interconnected either through the matrix porosity or through touching vugs [18]. Only a brief introduction to vuggy carbonate rock is given here. For a more comprehensive introduction consider carbonate reservoir textbooks, e.g. [18].

CT imaging of vuggy carbonate rocks is a nondestructive and noninvasive technique that can be used to obtain a digitized map of the carbonate minerals and pore space. The image data can be acquired at varying resolutions, which again yields varying degree of accuracy in the image volume. High-resolution μCT scans yields a detailed description of the vuggy porosity and also parts of the

matrix porosity. Depending on the resolution the amount of porosity found below image resolution will vary. For CT imaging each voxel is an average measurement of a small region. Thus, each voxel can contain information from several different materials. This is referred to as the *partial volume effect* [14]. It yields a blurring effect that becomes more visible for decreasing image resolution. Image object boundaries become less distinct, making it more and more difficult to distinguish between different objects in the image volume.

Other image artifacts, in addition to the partial volume effect, are also encountered in CT images. Two such artifacts are beam hardening and ringing effects. Beam hardening is due to an increase of mean X-ray energy as it passes through an object, which can lead to varying voxel intensity values for the same material. The ringing effect is often caused by faults in the detector and produces circular artifacts centered about the rotational axis [14, 37]. Furthermore, CT imaging generates a large amount of image data. This can lead to computer memory problems when processing the data, especially for 3D image data-sets. For more background information on CT imaging consider e.g. [31, 14, 37].

4.2 Carbonate rock image data-sets

The vuggy carbonate rock samples considered in this work were acquired from an outcrop of the Prebetic sub-zone, Betic range, Spain. A single block was removed, and further divided into several samples. Here we examine CT images of three different cylindrical core samples, acquired in 3D, of which selected slices can be observed in Figure 4.1. The core samples vary in size and so does the image resolution. All the image data-sets are truly cylindrical, which is ensured in a pre-processing step.

In Figure 4.1(a) data-set $\mathcal{A}$ is observed. It was obtained from the smallest core sample that measures approximately $5\,\mathrm{mm}$ in diameter and $5.5\,\mathrm{mm}$ in length. Of the data-sets it has the highest uniform resolution of $11.4\,\mu\mathrm{m}$ and therefore the most detail can be observed. The dark gray regions corresponds to vugs, while the lighter gray regions corresponds to two different rock materials. In its histogram, shown in (d), the three peaks indicate that the image phases are well separated. In addition to the three image phases, some high-intensity regions exists. These regions can lead to integer overflow that needs to be corrected in a pre-processing step. The data-set was obtained using a μCT scanner, which introduced some ringing artifacts in the first few slices.

In Figure 4.1(b) data-set $\mathcal{B}$ is observed. It was obtained using the same μCT scanner as the one used for data-set $\mathcal{A}$ and also here ringing artifacts has been introduced. This data-set is obtained from a larger core sample, which measured approximately $2.5\,\mathrm{cm}$ in diameter and $3\,\mathrm{cm}$ in length. It has uniform resolution

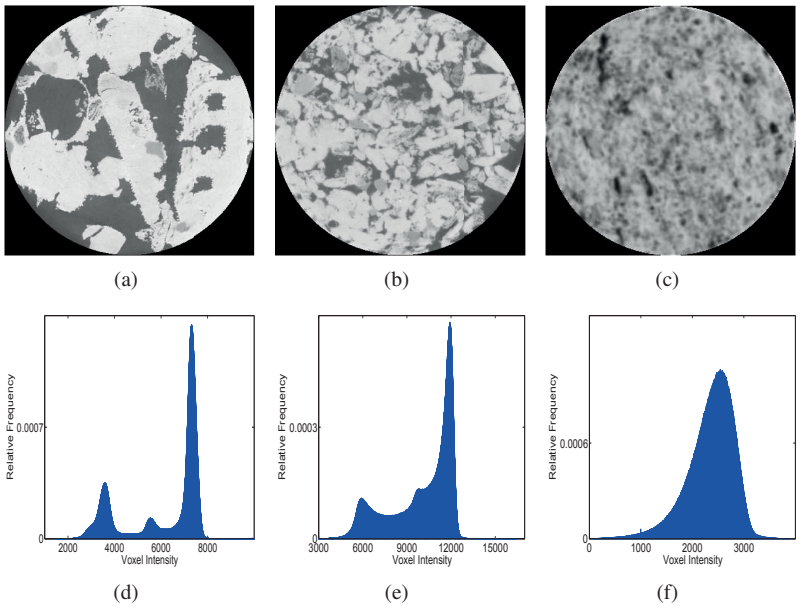


Figure 4.1: Three different 3D image data-sets are encountered. Data-set $\mathcal{A}$ is displayed in (a), which has the highest uniform resolution of $11.4\mu\mathrm{m}$ and therefore also displays the most detail. Data-set $\mathcal{B}$ is shown in (b). It is obtained from a larger core sample and has uniform resolution of $57.16\mu\mathrm{m}$. Data-set $\mathcal{C}$ is shown in (c), which is obtained from the largest core sample. It is blurry, which makes it challenging to differentiate between the pore space and rock matter. It has low non-uniform spatial resolution of $0.305 \times 0.305 \times 1.5\mathrm{mm}$. The histogram in (d) confirms that data-set $\mathcal{A}$ is detailed as it has three well separated histogram peaks. The one in (e) shown that less detail can be observed from data-set $\mathcal{B}$, although also here three different histogram peaks can be observed. The histogram in (f) clearly shows that data-set $\mathcal{C}$ is quite blurry as its histogram has the shape of a skewed gaussian. The background voxels in black are not included in the histograms.

of $57.16\,\mu\mathrm{m}$. Due to lower image resolution fewer details can be observed when compared to data-set $\mathcal{A}$. Even so, three different image phases are observed also here. They are visible in the image histogram as well, shown in (e), although the peaks are not as well separated as for data-set $\mathcal{A}$. This is due to lower image resolution that makes the partial volume effect greater. Also for this data-set a fourth image phase can be observed in a few locations containing high intensity voxel values.

In Figure 4.1(c) data-set $\mathcal{C}$ is shown. It is of the larges core sample that measures approximately $10\,\mathrm{cm}$ in diameter and $21\,\mathrm{cm}$ in length. This data-set is scanned using a different CT scanner than the two previous data-sets, and it has low non-uniform spatial resolution of $0.305 \times 0.305 \times 1.5\mathrm{mm}$. Due to its low resolution the partial volume effect is large and as a result the data is blurry. It is therefore more challenging to determine the pore space. This is also confirmed by its image histogram, shown in (f), which has the shape of a skewed gaussian. Thus, it is not possible to determine image phases directly from the histogram.

Of the three data-sets, data-set $\mathcal{A}$ is the easiest data-set to analyze. As the image resolution decreases image object boundaries become more and more blurry. Although data-set $\mathcal{B}$ also has high resolution, it can be difficult to distinguish between different image phases for this data-set. This makes data-set $\mathcal{B}$ more challenging to analyze than $\mathcal{A}$. Since data-set $\mathcal{C}$ has low non-uniform image resolution it can be difficult to determine image objects, which makes this data-set the most challenging to analyze. Data-sets $\mathcal{A}$ and $\mathcal{B}$ are considered to be three-phase. The fourth phase containing high intensity regions is ignored as it is not visible in the image histogram. For data-set $\mathcal{C}$ it is difficult to determine the number of phases. Since it is acquired from the same block of rock as data-sets $\mathcal{A}$ and $\mathcal{B}$, it is considered to be three-phase as well.

4.3 Expected visible porosity

To compute the expected visible porosity, ϕ_v, of the data-sets the measured porosity, ϕ_m, from laboratory experiments is used. For data-set $\mathcal{B}$ the porosity has been measured to $\phi_m = 31.4\,\%$. For data-set $\mathcal{A}$ and $\mathcal{C}$ the porosity has not been measured. However, in [35] the average measured porosity of several samples acquired from the same block of rock was $29\,\%$. It is assume to be the approximate porosity of these two data-sets.

An approximate measure of ϕ_v can be computed by combining ϕ_m with pore size distributions obtained from nuclear magnetic resonance (NMR) experiments. Pore size distributions were computed by [35] where they measured the T_2 distributions of three different vuggy carbonate rock core samples that were approximately $3.8\,\mathrm{cm}$ in diameter and $5\,\mathrm{cm}$ in length. For these experiments the pore size distribution were not converted from μs to μm. Later in [8] the pore size distribution of two core samples were measured by combining diffusion and relaxation measurements. Their experiments were performed on two core samples of approximate diameter $2.5\,\mathrm{cm}$ and length $3\,\mathrm{cm}$, which were extracted from the same block of rock as our samples. They resulted in the discrete pore size distributions, D_1 and D_2, given in μm, that are are displayed in Figure 4.2. Each point represents the porosity proportion for a given pore size. Since these NMR

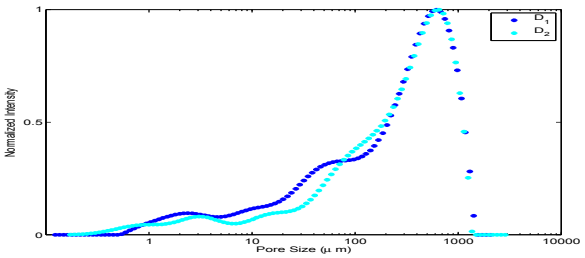


Figure 4.2: NMR pore size distributions of two data-sets. Each point represents the porosity proportion for a given pore size [8].

	Data-set $\mathcal{A}$	Data-set $\mathcal{B}$	Data-set $\mathcal{C}$
ϕ_v	$26.5\,\%$	$25\,\%$	$9\,\%$

Table 4.1: Computed expected visible porosity, ϕ_v. It yields an indication of how much porosity it is possible to detect in each data-set.

measurements have been converted from μs to μm our reasoning is based on their results.

To compute the CT images expected visible porosity, ϕ_v, the two distributions, D_1 and D_2, is used. ϕ_v is here computed by multiplying ϕ_m with the average portion of porosity above image resolution $R\,\mu$m. That is,

$$\phi_v = \phi_m \cdot \frac{1}{2} \sum_{x \geq R} \Big(D_1(x) + D_2(x) \Big) \,. \tag{4.1}$$

It yields an indication of how much porosity it is possible to detect in the image volumes. For data-set $\mathcal{A}$ $\phi_v = 26.5\,\%$, for data-set $\mathcal{B}$ $\phi_v = 25\,\%$ and for data-set $\mathcal{C}$ $\phi_v = 9\,\%$. These measurements are also given in Table 4.1. In the pore size distributions the uncertainty increase with the pore size. It might therefore be difficult to quantify the exact amount of larger vugs. This is of importance for data-set C where the image resolution, $R\,\mu$m, is located close to peak of the two distributions. In addition, the NMR experiments were performed on much smaller samples than data-set $\mathcal{C}$. The computed expected visible porosity for this data-set is therefore not as reliable and it is likely that it has been underestimated.

Chapter 5

Image segmentation techniques

There exists a vide variety of image segmentation techniques. In this chapter five different segmentation methods that are applied for segmenting the CT vuggy carbonate rock data-sets are presented. They separate an image I_0 into regions based on criteria that vary for each individual method. The three first methods are global thresholding methods, namely *basic global thresholding*, *Ridler and Calvard's* (RC) method and *Otsu's* method. The two remaining methods *thresholding by indicator kriging* (IK) and the *piecewise constant level set* (PCLS) method are more complex. In the following a brief introduction to the methods is given.

5.1 Basic global thresholding

Global image thresholding is the simplest way of performing image segmentation. Threshold values are often chosen after first evaluating the normalized histogram h of I_0, given by

$$h_i = \frac{1}{N} \sum (I_0 = i) \quad \text{for } i = 0, \ldots, L-1\,, \tag{5.1}$$

where N is the number of image voxels and L is the maximum voxel intensity. It yields the distribution of the voxel intensities and can be regarded as a probability distribution since it sums to 1. For two-phase image segmentation a single threshold T is required and the binary map I_b is then computed by

$$I_b = I_0 < T\,. \tag{5.2}$$

The threshold T is ideally selected so that it groups together voxels with close to uniform voxel intensity values. For multiphase image segmentation, multiple

thresholds T_j, $j = 1, \ldots, n-1$, are required and multiple binary maps I_b^j are computed,

$$I_b^1 = I_0 < T_1\,, I_b^2 = T_1 \leq I_0 < T_2\,, \ldots\,, I_b^n = T_{n-1} \leq I_0\,. \tag{5.3}$$

The resulting segmented image can then be represented by for instance

$$I_s = 1 \cdot I_b^1 + 2 \cdot I_b^2 + \ldots + n \cdot I_b^n\,, \tag{5.4}$$

where n is the number of image phases. A more thorough introduction to global image thresholding can be found in image processing textbooks such as [11].

Within petroleum research, the vuggy pore space is often determined by global thresholding. A single hard threshold T is often set either based on user preference or by using the measured porosity ϕ_m from laboratory experiments to compute the threshold. In the latter case the threshold is computed by,

$$T = \{t \,:\, \max \sum_{i=0}^{t} h_i \leq \phi_m\}\,, \quad \text{for } t \in [0,\, L-1]\,. \tag{5.5}$$

This approach does not take advantage of the image information nor the available image histogram information. It is therefore expected that better results can be obtained by taking the image data and histogram information into account. In this work threshold T is computed from the expected visible porosity, ϕ_v, from eq. (4.1) instead of ϕ_m.

5.2 The Ridler and Calvard's method

The Ridler and Calvard's (RC) method [25] is an iterative global thresholding method. It computes the optimally global threshold values by locating the midpoint between the means of two adjacent histogram peaks. These peaks indicate distinct group of voxels with similar intensity values. The computation terminates once the threshold values converge. In this work three-phase segmentation problems are mostly encountered. Thus, the method is presented for such problems and two thresholds are computed. They are iteratively computed by

$$\begin{array}{rclcrcl} T_1' & = & \lim_{k\to\infty} T_1^k, & \text{for } T_1^k & = & \frac{\mu_1^k + \mu_2^k}{2} \\ T_2' & = & \lim_{k\to\infty} T_2^k, & \text{for } T_2^k & = & \frac{\mu_2^k + \mu_3^k}{2} \end{array}\,. \tag{5.6}$$

For each iteration k the means $\{\mu_1^k,\, \mu_2^k\,, \mu_3^k\}$ of the three image phases are computed by

$$\mu_1^k = \frac{\sum_{i=0}^{T_1^k} i h_i}{\sum_{i=0}^{T_1^k} h_i}\,, \quad \mu_2^k = \frac{\sum_{i=T_1^k+1}^{T_2^k} i h_i}{\sum_{i=T_1^k+1}^{T_2^k} h_i}\,, \quad \text{and} \quad \mu_3^k = \frac{\sum_{i=T_2^k+1}^{L-1} i h_i}{\sum_{i=T_2^k+1}^{L-1} h_i}\,. \tag{5.7}$$

The above formulation can easily be extended to handle images with more phases or reduced in the case of two phase segmentation. The iterative scheme converges quickly and globally optimal threshold values are found.

For the vuggy carbonate rock data-sets the first threshold T_1' separates the pore space from the first rock phase, while the second threshold T_2' separates two different rock phases from each other. In this work, the goal is to obtain a good representation of the pore space. Thus, it is only the first threshold that is of interest. All voxels greater than T_1' can therefor be considered as part of rock matter, while the remaining voxels are considered as porous. The binary pore space is therefore given by

$$I_p = I_0 > T_1' \,. \tag{5.8}$$

5.3 Otsu's method

Otsu's method [23] is an iterative global image segmentation technique that computes the globally optimal thresholds by maximizing the between class variance. These classes can also be viewed as image phases. In this work three-phase image data-sets are encountered and therefore the three-phase Otsu's method is required. In the following, the generalized multiphase approach is presented.

The multiphase Otsu's method separates an image into a given set of classes, C_j, $j = 1, \ldots, K$, by finding the set of thresholds, $\{T_1', \ldots, T_{K-1}'\}$, that maximizes the between class variance,

$$\sigma_B^2(T_1', \ldots, T_{K-1}') = \max_{0<T_1'<\ldots<T_{K-1}'<L} \sigma_B^2(T_1, \ldots, T_{K-1}) \,. \tag{5.9}$$

This yields the set of globally optimal thresholds that best separate the image into image phases. They are obtained by statistical computations on the normalized image histogram, h. The general formula for σ_B^2 is given by

$$\sigma_B^2 = \sum_{j=1}^{K} P_j (\mu_j - \mu_G)^2 \,, \tag{5.10}$$

where P_j is the probability of occurrence of class C_j,

$$P_j = \sum_{i \in C_j} h_i \,, \quad \text{for } j = 1, \ldots, K \,, \tag{5.11}$$

and μ_j is the corresponding class mean. It is computed by

$$\begin{aligned}\mu_j &= \sum_{i\in C_j} iP(i|C_j) \\ &= \frac{1}{P_j}\sum_{i\in C_j} iP(i)P(C_j|i) \\ &= \frac{1}{P_j}\sum_{i\in C_j} iP(i)\,, \qquad \text{since } P(C_j|i)=1\,, \quad j=1,\ldots,K\,.\end{aligned} \tag{5.12}$$

The formulation for the class mean is the same formulation as the one used by the RC method in eq. (5.7), only using a generalized notation. To compute σ_B^2 an estimate of the global mean, μ_G, of the entire image

$$\mu_G = \sum_{i=0}^{L-1} ih_i \tag{5.13}$$

is also required. Thus, σ_B^2 is the sum of weighted variances of the class means, μ_j, and the global mean, μ_G. Both the probability of class occurrence P_j and class mean μ_j needs to be computed for all possible threshold combinations in order to find the combination that maximize the between class variance, σ_B^2. For the three-phase segmentation problems encountered in this work, the number of classes is $K = 3$ and two global thresholds $\{T_1', T_2'\}$ needs to be computed. This makes the maximization problem a 2D problem. The method is more computational expensive when compared to the two former methods. For more information consider e.g. [23, 11]. Also for Otsu's method the first threshold T_1' separates the pore space from the first rock phase while the second threshold T_2' separates the two rock phases from each other. Thus, the porosity map is also here computed according to eq. (5.8).

5.4 Thresholding by indicator kriging

The indicator kriging (IK) method was proposed by *Oh* and *Lindquist* in 1999 [20]. It introduced a local thresholding scheme for two-phase segmentation. The method segments an image into two populations, Π_0 and Π_1, by using a local spatial covariance estimate and IK, followed by a probability estimate. To solve the problem IK uses the *ordinary kriging* (OK) framework and the computations are performed on smoothed indicator functions. The method requires that a fraction of the voxels are a priori assigned to one of the two populations. The remaining voxels are left to be assigned by IK.

5.4.1 Introduction to ordinary kriging

The OK method provides a method for estimating the unknown value in one location from neighboring data points using a statistical model. It estimates the unknown value, $z(x_0)$, as an outcome of a random variable (RV), $Z(x_0)$, in location x_0 by estimating $\tilde{Z}(x_0)$ as the weighted linear combination of neighboring RVs, $Z(x_\alpha)$,

$$\tilde{Z}(x_0) = \sum_{\alpha=1}^{n} \lambda_\alpha Z(x_\alpha)\,. \tag{5.14}$$

The neighboring RVs, $Z(x_\alpha)$, have known outcomes, $z(x_\alpha)$, for $\alpha = 1, \ldots, n$, and λ_α are the linear weights [12, 20]. The $n+1$ neighboring data points make up the kriging window, K_w, which is centered on x_0 [20]. By assuming a circular window, K_w is given as a disk in 2D or a ball in 3D.

For the OK method the statistical model is assumed stationary. The univariate and bivariate probability laws are then assumed independent of the location within the domain. The RVs are characterized by a common but unknown expected value,

$$E\{Z(x_\alpha)\} = \mu_z\,, \quad \forall\, x_\alpha \in K_w\,, \tag{5.15}$$

which is valid for all RVs within the kriging window, K_w. The covariance function between pairs of RVs depends only on the distance $\mathbf{h}$ separating them. It is given by

$$Cov(Z(x_\alpha),\, Z(x_\beta)) = C(x_\alpha - x_\beta) = C(\mathbf{h})\,, \quad \forall\, x_\alpha,\, x_\beta \in K_w\,, \tag{5.16}$$

and is computed during the kriging operation. The covariance function between two RVs is in the following also denoted by $C_{\alpha\beta}$. It yields a measurement of how they change together. When the output is a large positive number the two RVs are similar and in the opposite case they are not alike. The covariance function is isotropic since it depends only on the distance, $\mathbf{h} = ||x_\alpha - x_\beta||$, between two RVs [12, 20]. The covariance is related to the semi-variogram $\gamma(\mathbf{h})$, which can be expressed as

$$C(\mathbf{h}) = c - \gamma(\mathbf{h})\,, \tag{5.17}$$

where c is a constant that remains fixed throughout the computations. The semi-variogram is a measurement of the variance of the difference between two RVs, $Z(x_\alpha)$ and $Z(X_\beta)$. To ensure that the OK method yields an unique solution the semi-variogram needs to be modeled [6, 12, 13]. For more information on the covariance function and its relation to the semi-variogram, as well as modeling of the semi-variogram, consider e.g. [6, 12, 13].

Minimum variance unbiased estimator

The OK method is a minimum variance unbiased estimator. It is unbiased since it requires that the modeled error RV, $R(x_0) = Z(x_0) - \tilde{Z}(x_0)$, is equal to zero,

$$E\{R(x_0)\} = E\{Z(x_0) - \tilde{Z}(x_0)\} = 0\,. \tag{5.18}$$

It can be shown that this requirement is satisfied as long as the estimator satisfies the constraint,

$$\sum_{\alpha=1}^{n} \lambda_\alpha = 1\,. \tag{5.19}$$

This constraint is referred to as the linear constraint since it impose a constraint on the linear weights λ_α for $\alpha = 1, \ldots, n$ [12]. The OK method is a minimum variance estimator since it minimize the variance of the error RV, $R(x_0)$. To do so an expression for the error variance, Var$\{R(x_0)\}$, is needed. It can be expressed as

$$\text{Var}\{R(x_0)\} = \text{Var}\{Z(x_0)\} + \sum_{\alpha=1}^{n}\sum_{\beta=1}^{n} \lambda_\alpha \lambda_\beta C_{\alpha\beta} - 2\sum_{\alpha=1}^{n} \lambda_\alpha C_{\alpha 0}\,. \tag{5.20}$$

To ensure that the estimator is unbiassed the linear constraint is included in the expression for the error variance by adding an additional term,

$$\begin{aligned} \text{Var}\{R(x_0)\} \;\; = \;\; & \text{Var}\{Z(x_0)\} + \sum_{\alpha=1}^{n}\sum_{\beta=1}^{n} \lambda_\alpha \lambda_\beta C_{\alpha\beta} \\ & -2\sum_{\alpha=1}^{n} \lambda_\alpha C_{\alpha 0} + 2\nu\Big(\sum_{\alpha=1}^{n} \lambda_\alpha - 1\Big)\,. \end{aligned} \tag{5.21}$$

The introduced term ensures that the error variance is minimized while naturally satisfying the constraint. Since the linear weights λ_α sum to 1 the introduced term does not change the equality. The error variance is then minimized by setting its partial derivatives to zero [12]. A more detailed introduction to the estimator and derivation of the partial derivatives can be found in [12].

Ordinary kriging system

The *ordinary kriging* (OK) *system* is obtained by setting the partial derivatives to zero, which yields $n + 1$ equations,

$$\begin{matrix} \sum_{\beta=1}^{n} \lambda_\beta C_{\alpha\beta} + \nu = C_{\alpha 0} \\ \sum_{\beta=1}^{n} \lambda_\beta = 1 \end{matrix} \quad , \quad \text{for } \alpha = 1, \ldots, n\,. \tag{5.22}$$

Solving the linear system yields the vector that minimize the error variance, Var$\{R(x_0)\}$, from eq. (5.21). The linear system might produce negative weights that should be corrected by setting them to zero as they can produce negative estimates. The system has a unique solution as long as it is non-singular and the uniqueness is ensured through modeling of the semi-variogram [12]. Consider [12] for more detailed information.

5.4.2 Image segmentation by indicator kriging

The IK method is a two step segmentation procedure. In the first step a portion of the voxels are assigned to one of the two populations, Π_0 and Π_1. In the second step the remaining voxels are assigned by IK. This makes the method a local thresholding method. By using indicator functions the IK problem can be solved using the framework of the OK method [20].

A priori voxel assignment

In the first step of the IK method a portion of the voxels are assigned to either population Π_0 or Π_1. This initial segmentation of the data is performed by assigning all voxels below threshold T_0 to population Π_0 and all voxels above threshold T_1 to population Π_1. Thus we have

$$\Pi_0 = \{x \,:\, I_0(x) \leq T_0\} \tag{5.23}$$

and

$$\Pi_1 = \{x \,:\, I_0(x) \geq T_1\}\,. \tag{5.24}$$

The two thresholds can be determined from the image histogram of the original data, either by a trained user or based on some method. The remaining undetermined voxels,

$$x \notin \{\Pi_0 \cup \Pi_1\}\,, \tag{5.25}$$

are assigned by IK. These voxels mainly lies along the image object boundaries separating Π_0 and Π_1 [20].

Segmentation by indicator kriging

In the second step the IK method assigns the remaining undetermined voxels to either population Π_0 or Π_1 by computing the probability that a voxel x_0 belongs to one of the two populations using the OK framework. The IK method requires that indicator functions are computed from the image data. The OK system is then solved for indicator covariance functions prior to estimating the probability that voxel x_0 belongs to either Π_0 or Π_1.

Indicator functions

Indicator functions are used to indicate whether or not a voxel is a member of a given population. For the IK method smoothed indicator functions are used in which voxels are given an indicator value in range $[0,\ 1]$. They are defined as

$$\hat{i}(T_i;\, x) = \begin{cases} 1\,, & \text{if } I_0(x) < T_i - s_i^l \\ 0\,, & \text{if } I_0(x) > T_i + s_i^r \\ \frac{F(T_i+s_i^r)-F(I_0(x))}{F(T_i+s_i^r)-F(T_i-s_i^l)}\,, & \text{otherwise} \end{cases}, \quad \text{for } i = 0, 1\,. \tag{5.26}$$

The T_i's are the initial thresholds from the a priori assignment step and $F(\cdot)$ is the cumulative distribution function,

$$F(l) = Pr\{I_0(x) \leq l\}\,, \quad \text{for } l = \min\{I_0\}, \ldots, \max\{I_0\}\,. \tag{5.27}$$

The parameters s_0^l, s_0^r, s_1^l and s_1^r are defined as $s_0^l \equiv s_1^r \equiv 0$ and $s_0^r = s_1^l = \frac{(\sigma_0 T_1 + \sigma_1 T_0)}{(\sigma_0+\sigma_1)}$, where σ_i is the sample standard deviation of population Π_i,

$$\sigma_i = \left(\frac{1}{N_i - 1} \sum_{j=0}^{N_i-1} \left(I_0(x_j) - \mu\right)^2 \right)^{\frac{1}{2}}, \quad \text{for } \mu = \frac{1}{N_i} \sum_{j=0}^{N_i-1} I_0(x_j)\,, \tag{5.28}$$

where $x_j \in \Pi_i$ and N_i is the number of voxels in population Π_i, $i = 0, 1$ [20]. The smoothed indicator functions are indicative of how likely it is that a voxel x belongs to one of the two populations. All voxels given indicator value 1 are indicated to belong to population Π_0. Similarly, all voxels given indicator value 0 are indicated to belong to population Π_1. The remaining voxels lies in a transition region with indicator values in range $\langle 0,\ 1 \rangle$.

Solving the indicator kriging method

Once the indicator functions are computed the IK problem can be solved by solving the OK linear system in eq. (5.22) for indicator covariance functions. The linear system now becomes

$$\begin{matrix} \sum_{\beta=1}^{n} \lambda_\beta^i C_{I\alpha\beta}^i + \nu^i = C_{I\alpha 0}^i \\ \sum_{\beta=1}^{n} \lambda_\beta^i = 1 \end{matrix}, \quad \text{for } \alpha = 1, \ldots, n\,,\ i = 0, 1\,, \tag{5.29}$$

where the covariance function $C_{I\alpha\beta}^i$ is computed from the corresponding indicator function. To ensure that the OK system yields an unique solution the semivariogram needs to be modeled. For a more detailed description of the semivariogram consider e.g. [12].

The linear system might produce negative weights that should be adjusted. For IK the weights λ_α^i are interpreted as probabilities. Thus, negative weights can produce negative estimates [20]. In [20] they followed the procedure proposed in [7] in which small positive weights are set to zero in addition to the negative weights as they are the result of oscillating negative weights. The positive weights are set to zero if they are smaller than the average magnitude of the negative weights,

$$\bar{\lambda}^i = \frac{1}{n'} \sum_{\gamma=1}^{n'} |\lambda_{x_\gamma}^i| \,, \quad \text{where } \lambda_{x_\gamma}^i < 0 \text{ for } \gamma = 1, \ldots, n' \,, \tag{5.30}$$

and if their corresponding covariance $C_{I\alpha 0}^i$ is less than the average covariance of the negative weights,

$$\bar{C}^i = \frac{1}{n'} \sum_{\gamma=1}^{n'} C_{Ix_\gamma 0}^i \,. \tag{5.31}$$

Probability estimate

Once the negative weights have been adjusted, the segmentation is completed by computing the probability that the undetermined voxel x_0 belongs to either Π_0 or Π_1. The probability estimate is expressed in terms of the linear weights and indicator functions by

$$\begin{aligned} P(T_i;\, x_0|n) &= Pr\{I_0(x_0) \leq T_i\} \\ &= \sum_{\alpha=1}^{n} \lambda_\alpha^i \; \hat{i}(T_i;\, x_\alpha) \,, \quad \text{for } \lambda_\alpha^i \geq 0 \,, \;\; i = 0,\, 1. \end{aligned} \tag{5.32}$$

The probability that a voxel belongs to population Π_0 is given by $P(T_0;\, x_0|n)$ whereas the probability that it belongs to population Π_1 is $1 - P(T_1;\, x_0|n)$. The undetermined voxel is assigned to the population with the largest probability,

$$Z(x_0) \in \begin{cases} \Pi_0 \,, & \text{if } P(T_0;\, x_0|n) > 1 - P(T_1;\, x_0|n) \\ \Pi_1 \,, & \text{otherwise} \end{cases} \,. \tag{5.33}$$

This completes the IK method. The computations are repeated for all the undetermined voxels [20].

Majority filtering

During the a priori assignment step a certain number of voxels might be misclassified. To limit this problem, majority filtering (MF) was included in [20] as a

way to remove isolated voxels in the a priori assignment map. It is applied only to those voxels a priori assigned to populations Π_0 or Π_1.

MF is a windowing method that change the population type of a voxel if it is surrounded by voxels mostly belonging to the opposite population. The corresponding entries in the indicator functions need also be updated. If a voxel x_0 is changed to Π_0 then the indicator functions are reset to $\hat{i}(T_0;\, x_0) = \hat{i}(T_1;\, x_0) = 1$, and similarly if it is changed to Π_1 they are reset to $\hat{i}(T_0;\, x_0) = \hat{i}(T_1;\, x_0) = 0$ [20].

Once the segmentation process has been completed, a second pass of MF can be applied. Also now the method is only applied to those voxels a priori assigned and the population type is changed only if the voxel is surrounded by voxels mostly belonging to the opposite population. For both MF passes a voxel belonging to one population is said to belong to the opposite population if at least 60% of its neighbors belong to it [20].

5.4.3 Method modification for vuggy carbonate rock images

The original IK segmentation method is valid for two-phase image data-sets. The CT images of vuggy carbonate rock encountered in this work are, however, three-phase. Using the original IK formulation the first rock phase has a tendency to be wrongly classified. To obtain reasonable segmentation results the method is slightly modified.

A priori assignment step

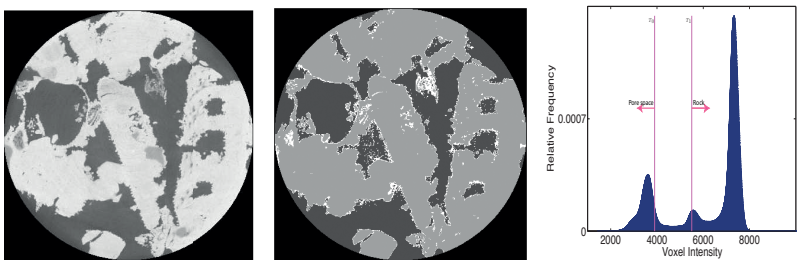
The a priori voxel assignment step requires two thresholds, T_0 and T_1, to a priori assign a portion of the voxels to Π_0 or Π_1. In this work the thresholds are computed using a two step procedure. In the first step the Ridler and Calvard's (RC) method [25], introduced in Section 5.2, is employed. Two thresholds, T_1' and T_2', are computed that separate the image phases since the vuggy data-sets contain three different image phases. In the second step the two thresholds, T_0 and T_1, required by the IK method are computed from T_1' since it separates the pore space from rock matter. They are found by

$$T_0 = T_1' - T_1' \cdot f_0\,, \quad \text{for } f_0 \in \mathbb{Z}^+\,, \tag{5.34}$$

and

$$T_1 = T_1' + T_1' \cdot f_1\,, \quad \text{for } f_1 \in \mathbb{Z}^+\,, \tag{5.35}$$

where f_0 and f_1 are user defined. Given T_0 and T_1 the a priori population assignment can be performed. The remaining voxels are assigned by IK.



(a) (b) (c)

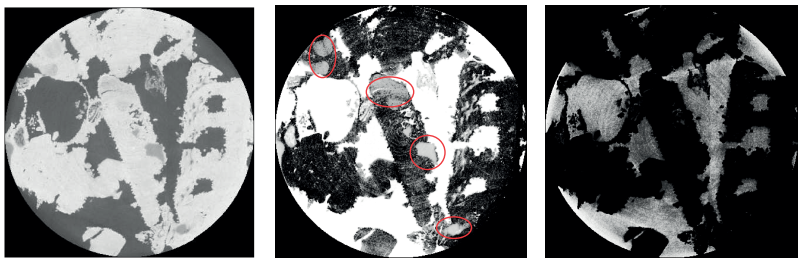
Figure 5.1: A priori population assignment for data-set $\mathcal{A}$. In (a), an input image slice is shown and in (b), the computed a priori assignment map. All voxels in gray are assigned to population Π_0, all light gray voxels are assigned to Π_1 and the remaining voxels in white are left undetermined. In (c), the two thresholds used in the a priori assignment step is shown on the normalized image histogram. The arrows indicate the parts of the histogram a priori assigned to Π_0 and Π_1.

In Figure 5.1 an illustration of this process is given for data-set $\mathcal{A}$. In (a) an input image slice is shown and in (b) the resulting partitioning after the a priori assignment process has been completed. All the voxels assigned to Π_0 (pore space) are shown in dark gray and all the voxels assigned to Π_1 (rock) are shown in light gray. The voxels in white are the undetermined voxels that mostly lies along the image object boundaries. In 5.1(c) the normalized image histogram of the input image is shown on which the two thresholds, T_0 and T_1, are indicated. The arrows indicate the portions of the histogram a priori assigned to Π_0 and Π_1. Since it is only the boundary between the pore space and rock that is of interest, the second and third image phases are regarded as a single phase.

Adaptation of the indicator function

The smoothed indicator function, $\hat{i}(T_i;\, x)$ for $i = 0,\, 1$, from eq. (5.26) is indicative of how likely it is that a voxel belongs to one of the two populations, Π_0 and Π_1. It can be applied for two-phase image data-sets. Our data-sets are, however, three-phase. This have to be taken into account when computing the indicator functions.

One option is to treat the two rock phases as a single image phase. The standard deviation σ_1 will then, however, become quite large. This influences in particular indicator function $\hat{i}(T_0;\, x)$, which will have a tendency to wrongly indicate the first rock phase as porous. This can be seen in Figure 5.2. By comparing the input image in (a) to the first indicator function $\hat{i}(T_0;\, x)$ in (b) it is observed that



(a) (b) (c)

Figure 5.2: Problem original smoothed indicator functions. In (a) an input image slice of data-set $\mathcal{A}$ is shown, and in (b) and (c) the corresponding indicator functions $\hat{i}(T_0;\, x)$ and $\hat{i}(T_1;\, x)$, respectively. Observe in (b) that the encircled regions are wrongly indicated to belong to population Π_0. This problem is not observed for $\hat{i}(T_1;\, x)$ shown in (c).

some regions, such as the encircled ones, are wrongly indicated to be closer to porous phase than the rock phase. As a result, IK have a tendency to wrongly assign this rock phase to Π_0 instead of Π_1 during the IK segmentation process. Due to the formulation of the indicator function $\hat{i}(T_1;\, x)$ in (c) is not noticeably influenced.

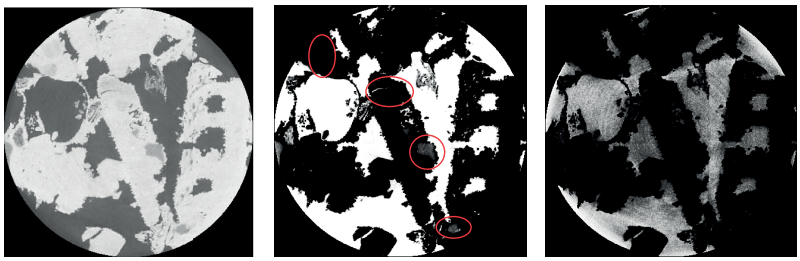
Another option is to adapt the indicator function slightly by taking into account that the objective is to locate the boundary between the pore space and rock matter. The intersection between the two rock phases is not of interest. Therefore, all voxels satisfying $I_0 > T_2'$ are said to be known as part of population Π_1. This leads to the modified smoothed indicator functions,

$$\hat{i}(T_i;\, x) = \begin{cases} 1\,, & \text{if } I_0(x) < \max\{\min\{I_0\}, T_i - s_i^l\} \\ 0\,, & \text{if } I_0(x) > \min\{T_2'\,,\, T_i + s_i^r\} \\ \frac{F_u - F(I_0(x))}{F_u - F_l}\,, & \text{otherwise} \end{cases}\,, \tag{5.36}$$

for $i = 0, 1$. All voxels greater than T_2' are set to 0, indicating that they are part of population Π_1. The parameters s_i^l and s_i^r are as previously given by $s_0^l \equiv s_1^r \equiv 0$ and $s_0^r = s_1^l = \frac{\sigma_0 T_1 + \sigma_1 T_0}{\sigma_0 + \sigma_1}$, and the first standard deviation σ_0 is computed from population Π_0 as before. The second standard deviation σ_1 is, however, now computed for all voxels satisfying $x \in \{\Pi_1 \cap \{x \,:\, I_0(x) \leq T_2'\}\}$. The cumulative distribution function has also been modified,

$$F(l) = Pr\{I_0(x) \leq l\}\,, \quad \text{for } l = \min\{I_0\}, \ldots, T_2'\,. \tag{5.37}$$

It is only computed until T_2' is reached since all the voxels greater than T_2' are defined to be part of population Π_1. In the transition function, the upper limit F_u



(a) (b) (c)

Figure 5.3: Modified indicator functions. In (a) an input image slice of data-set $\mathcal{A}$ is shown, and in (b) and (c) the corresponding adapted indicator functions $\hat{i}(T_0;\, x)$ and $\hat{i}(T_1;\, x)$, respectively. The encircled regions in $\hat{i}(T_0;\, x)$ corresponding to the first rock phase are now more correctly indicated to belong to population Π_1.

and lower limit F_l are given by

$$F_u = F(\min\{T_2', T_i + s_i^r\}) \tag{5.38}$$

and

$$F_l = F(\max\{\min\{I_0\}, T_i - s_i^l\}). \tag{5.39}$$

This yields improved smoothed indicator functions that better indicates whether or not a voxels belongs to Π_0 or Π_1 as can be observed in Figure 5.3. In (b) it is seen that indicator function $\hat{i}(T_0;\, x)$ now more correctly indicate that the first rock phase belongs to population Π_1. The indicator function $\hat{i}(T_1;\, x)$ in (c) does not appear to be influenced by the adaptation.

5.5 The piecewise constant level set method

The piecewise constant level set (PCLS) method was first proposed by *Lie et al.* in 2006 [15] as a region based image segmentation method for multiphase problems. It segments an image by minimizing an energy functional. Traditionally this is performed by solving an *Euler-Lagrange* equation, which is time consuming. In this work the minimization problem is instead solved by computing the *minimum cut* on a specially designed graph G. This *graph cut* solver was first introduced in [2] for 2D problems. In Paper D it was extended for handling 3D problems.

5.5.1 Introduction to the piecewise constant level set method

The *level set method* was introduced by *Osher* and *Sethian* in 1988 [22]. It is an active contour model that can be applied for image segmentation. The method represents evolving fronts by embedding them in an higher dimensional function, which is usually given as a signed distance function. This function is commonly referred to as the level set function, φ. The evolving fronts are usually represented by the zero level set, which allows for representing complicated structures that do not depend on the discretization. Topological changes in the fronts, such as splitting and merging, are naturally handled. The evolving fronts propagates under forces in the normal direction of φ itself until (ideally) reaching steady state. This steady state solution corresponds to the segmentation. To ensure that the problem remains well-posed reinitialization of the level set function is required [21].

The *Chan-Vese* model [5] used the level set framework to solve the *Mumford-Shah* model [19] for two-phase image segmentation problems. It is a region based segmentation method that can detect image objects both with and without gradient information. The method was later extended for handling multiphase segmentation problems in [34] by using several φ's to represent the evolving fronts. Traditionally, the methods based on the level set approach are solved by first deriving an Euler-Lagrange equation and then solving the problem using standard numerical schemes.

The PCLS method [15] was later introduced as an alternative region based segmentation approach for multiphase problems. It approximates an image I_0 by a piecewise constant image function $\tilde{I}$ using a single PCLS function φ, which is defined over the whole image domain Ω. In the original PCLS formulation the *Mumford-Shah* energy functional was reformulated as

$$E(\mathbf{c}, \varphi) = \int_{\Omega} (I_0 - \tilde{I}(\varphi))^2 d\bar{x} + \frac{\nu}{2} \sum_{i=1}^{n} \int_{\Omega} |\nabla \psi_i(\varphi)| d\bar{x} \,. \tag{5.40}$$

It is minimized with respect to the mean phase values $\mathbf{c}$ and PCLS function φ. For an n-phase segmentation problem Ω is partitioned into n subdomains $\{\Omega_i\}_{i=1}^{n}$ and the PCLS function is defined as $\varphi = i \in \Omega_i$, for $i = 1, \ldots, n$. The energy functional also makes use of characteristic functions ψ_i, which are equal to 1 for domain Ω_i and 0 otherwise. Similar results can also be obtained by regularizing ϕ directly (see e.g. [32]). The PCLS energy functional then becomes

$$E(\mathbf{c}, \varphi) = \int_{\Omega} (I_0 - \tilde{I}(\varphi))^2 \, d\bar{x} + \nu \int_{\Omega} |\nabla \varphi| \, d\bar{x} \,. \tag{5.41}$$

This is the formulation used in this work. The first term is minimized when $\tilde{I}$ approximates I_0 and its discontinuities represents image object boundaries. These

discontinuities are modeled by the discontinuities of φ. The second term impose regularization on φ. It ensures that the boundaries remains smooth and that small objects are removed. Its influence is controlled by the parameter ν. The larger ν is, the smoother the resulting segmentation will be. The mean phase values $\mathbf{c} = \{c_1, \ldots, c_n\}$ are computed by

$$c_i = \frac{\int_\Omega I_0 \psi_i(\varphi)\, d\bar{x}}{\int_\Omega \psi_i(\varphi)\, d\bar{x}}\,, \quad \forall\, i = 1, \ldots, n\,, \tag{5.42}$$

and the piecewise constant image function is found by

$$\tilde{I}(\varphi) = \sum_{i=1}^{n} c_i \psi_i(\varphi)\,, \tag{5.43}$$

for a fixed φ. The PCLS function need not be reinitialized as it is represented as a piecewise constant function. Consider Paper D for more details.

5.5.2 The piecewise constant level set method solved by graph cut

Fast minimization of the PCLS energy functional from eq. (5.41) can be obtained through representing the problem on a special graph G and computing the *minimum cut* on this graph. It was suggested in [2] to discretize the PCLS energy functional directly to obtain a graph representable minimization problem. The discretization yields

$$E_d(\mathbf{c},\, \varphi) = \sum_{p \in D} (I_{0_p} - \tilde{I}_p)^2 \Delta + \nu \mathrm{TV}_d(\varphi)\,, \tag{5.44}$$

where p corresponds to voxels, domain D has the same dimensionality as I_0 and $\Delta = \prod_{i=1}^{3} \Delta_i$. The grid spacing in direction i is Δ_i. The regularization term is now given by the discrete *total variation* (TV) of φ, which is equal to the L_1-norm of the gradient. It was shown in [26] for denoising that using the L_1 norm preserves image edges while removing noisy artifacts. The regularization term can then be represented on discrete form as

$$TV_d(\phi) = \sum_{p \in D} \sum_{q \in N f_\kappa(p)} \frac{1}{2} \omega_{pq} |\phi_p - \phi_q|, \qquad \text{for } \omega_{pq} = \frac{\Delta}{\|p - q\|_2}\,. \tag{5.45}$$

The edge weight ω_{pq} between voxel p and q is divided by 2 since $\omega_{pq} = \omega_{qp}$. Voxel q lies in the neighborhood of p and for 3D problems a 6- or 26-neighborhood system is applied, which are denoted N_6 or N_{26}, respectively.

Once the graph representable energy functional is obtained, a special graph $G = (V, E)$ can be constructed. It consists of a set of vertices V and a set of directed edges E connecting the vertices. The idea of [2] was to create a multi-level graph in order to solve multiphase segmentation problems. For 3D problems, a multi-volume graph needs to be constructed. The set of vertices V can then be defined as

$$V = \{v_{p,l} \,|\, \forall p \in D,\, l \in \{1, \ldots, n-1\}\} \cup \{v_s\} \cup \{v_t\} \tag{5.46}$$

for a n-phase problem. Thus, the graph has $n-1$ levels and at each level there are vertices $v_{p,l}$ corresponding to image voxels, in addition to the terminal nodes v_s and v_t. The terminal nodes are used for partitioning V into two disjoint sets and the levels are used to further partition graph into n sets. The total set of edges E is given by

$$E = E_D \cup E_R\,, \tag{5.47}$$

where E_D are the between volume edge set and E_R are the directed within volume edge set. To each edge e weights ω_e are assigned that are derived from eq. (5.44). From the data-term the between volume edge weights are derived, given by

$$\begin{array}{lll} \omega_e(v_s, v_{p,1}) & = (u^0(p) - c_1)^2\Delta, & \forall p \in D \\ \omega_e(v_{p,l}, v_{p,l+1}) & = (u^0(p) - c_{l+1})^2\Delta, & \forall p \in D,\, l = \{1, \ldots, n-2\}\,. \\ \omega_e(v_{p,n-1}, v_t) & = (u^0(p) - c_n)^2\Delta, & \forall p \in D \end{array} \tag{5.48}$$

From the regularization term the within volume edge weights are derived. They are given by

$$\omega_e(v_{p,l}, v_{q,l}) = \omega_{pq}, \quad \forall p \in D,\, q \in N_{\kappa=6,26},\, l = \{1, \ldots, n-1\}\,, \tag{5.49}$$

where ω_{pq} is as defined in (5.45). These weights do not depend on the graph level.

Once the graph has been constructed and the weights are assigned, the graph cut problem can be solved. Such problems are solved by computing the *minimum cut* on the G, which is equivalent to computing the *maximum flow* due to the duality theorem in [10]. A *cut* C on graph G is the edge set, $C \subset E$, that separates the vertices into two disjoint sets, V_s and V_t, so that $v_s \in V_s$ and $v_t \in V_t$. For multi-volume graphs the levels are used to further differentiate between the different phases. The cost of the cut is computed as the sum over all edges $e \in C$,

$$|C| = \sum_{e \in C} \omega_e\,. \tag{5.50}$$

The minimum cut is the cut with the smallest total cost, which is also the cut that minimize the energy functional [4]. For the PCLS energy functional (5.41) the minimum cut is found by minimizing $E_d(\mathbf{c},\, \varphi)$ in eq. (5.44) by

$$\min_{C\ on\ G} |C| = \min_{\phi} \sum_{p\in D} (u_p^0 - \tilde{u}_p)^2 \Delta + \nu \sum_{p\in D} \sum_{q\in N_\kappa(p)} \frac{1}{2}\omega_{pq} |\phi_p - \phi_q| \,. \qquad (5.51)$$

This is the cut that minimize the PCLS energy functional. For the cut to be admissible it should sever at least on edge $e \in E_D$ for each vertex $p \in D$, otherwise it does not separate v_s and v_t. If it severs more than one edge it is not minimum cut since severing a single edge will always lead to a smaller total cost [2]. The minimization problem is solved by *maximum flow* based on the work of [4].

From the cut C the PCLS function φ can be constructed by taking into account the level in graph G of the edges in the cut, $e \in C$. For an n-phase problem φ can be defined as

$$\varphi(p) = \begin{cases} 1, & \text{if } (v_s, v_{p,1}) \in C \\ l+1, & \text{if } (v_{p,l}, v_{p,l+1}) \in C \,, \\ n, & \text{if } (v_{p,n-1}, v_t) \in C \end{cases} \qquad l = \{1, \ldots, n-2\} \,. \qquad (5.52)$$

Its discontinuities corresponds to the segmentation [2]. Observe that φ is not required for initiating eq. (5.48), it is sufficient to provide an initial guess of the mean phase values. φ can be extracted as needed during the computations. The mean phase values $\mathbf{c}$ and the approximate image function $\tilde{I}$ are computed for a fixed φ. The number of phases can be overdetermined and some phases can be empty in the result. Consider Paper D for more details about the graph cut approach.

In Figure 5.4 the principle of the special graph construction is presented for simple 2D examples. In (a) an ordinary graph construction applied for two-phase image segmentation problems is shown. There it is observed that the graph is separated into two parts by the cut C. The planar edges corresponds to E_R and the vertical edges that are connected to the two terminal nodes to E_D. For illustration purposes only edges in the cut are shown for E_D . In (b) the image segmentation problem is three-phase. A graph with two levels is therefore required. There the cut C severs the graph into three parts by using the between level edges to further partitioning the graph. In (c) an illustration is given of how the PCLS function φ is constructed. Observe that it is constructed by taking into account the edges of E_D that are in the cut C and their level in the G. The resulting segmentations are shown beneath the graphs.

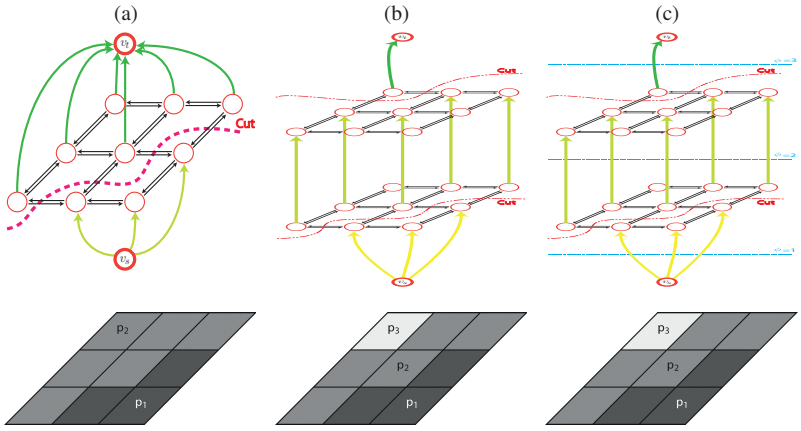


Figure 5.4: Illustration of 2D graph construction. In (a) the segmentation problem is two-phase and therefore a single level is required in G. The cut C partitions the graph into two parts. In (b) the problem is three-phase. Thus, two levels are needed to represent the problem and the between level edges are used to further partition the graph. In (c) the PCLS function φ is constructed by taking into account the between level edges in the cut C and their level in G. For illustration purposes only the edges in the cut are shown for the between level edge set E_D. The resulting segmentations are shown below the graphs.

Chapter 6

Results and conclusion

In this chapter a summary of the segmentation results from applying the previously presented segmentation techniques to CT imaged vuggy carbonate rock is given. The level of difficulty varies between the image data-sets as they have varying image resolutions. For the two μCT data-sets, where image details are more visible, the segmentation methods also succeed in a larger degree in capturing the true complexity of the image data. As the image resolution decrease it becomes increasingly difficult to differentiate between pore-space and rock matter, and as a result the accuracy in the segmentation decrease as well. For a more thorough coverage the reader is referred to Paper B and Paper C. The latter paper is an extension of the former.

6.1 Summary of the segmentation results

In this section a summary of the visual evaluation of the segmentation results is given. The segmentation methods have been applied to data-set $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ from Section 4.2, and an image slice for these data-sets is shown in Figure 6.1 (a)-(c), respectively. Data-set $\mathcal{A}$ has the highest uniform image resolution of $11.4\,\mu\mathrm{m}$ and $\mathcal{B}$ has uniform resolution of $57.16\,\mu\mathrm{m}$. Data-set $\mathcal{C}$ has low non-uniform resolution of $0.305 \times 0.305 \times 1.5\,\mathrm{mm}$, which makes its image objects blurry. The two first data-sets were acquired using the same μCT scanner, while the last data-set was acquired using a different CT scanner.

Basic global thresholding For basic global thresholding threshold T was computed from the expected visible porosity, ϕ_v. The computations show that for data-sets $\mathcal{A}$ and $\mathcal{B}$ the approach detects a too large a portion of the voxels as porous. For data-set $\mathcal{C}$ it appears to both under- and overestimate the pore space due to the highly blurred data, which yields strongly overlapping image phases in

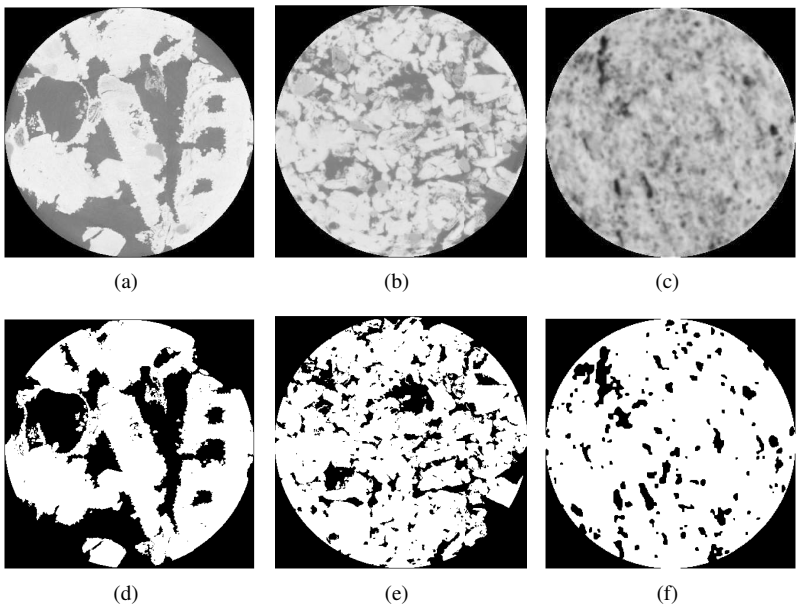


Figure 6.1: On the *upper row* selected slices of the three encountered data-sets are shown. Data-set $\mathcal{A}$ is shown in (a), data-set $\mathcal{B}$ in (b), and Data-set $\mathcal{C}$ in (c). On the *lower row* the corresponding segmentation using the PCLS method is displayed.

the histogram. For this data-set ϕ_v is probably underestimated as was mentioned in Section 4.3.

Computing the threshold solely based on ϕ_v does not take the available image nor histogram information into account. It is therefore natural to expected that a more complex approach will provide an improvement. An advantage of this approach is that it is fast to compute.

The RC method The RC method computes the optimal global thresholds as the midpoint between the means of adjacent histogram peaks. The method is fast to compute and the solution is found within seconds.

It is observed that the RC method describes the pore space well for both data-sets $\mathcal{A}$ and $\mathcal{B}$. For data-set $\mathcal{B}$ the method does not, however, capture narrow valleys of porosity as well as it does for data-set $\mathcal{A}$. This is due to lower image resolution, which makes the narrow channels more blurry. In the pore space of both segmentation results a vast number of small pore objects are detected. It can be difficult to determine if these objects correspond to true porosity or if they have

been wrongly classified. Compared to basic global thresholding based on ϕ_v the RC method detects slightly less porosity, which is expected since it computes a globally optimal threshold from the image histogram. It does not capture overlapping histogram populations. Thus, it should detect less porosity than the expected visible porosity, ϕ_v.

For data-set $\mathcal{C}$ a larger percentage of porosity is detected when compared to basic global thresholding. This was expected since ϕ_v is probably underestimated for this data-set. Due to the low image resolution global thresholding methods will never provide an accurate segmentation since the image phases strongly overlap. For this data-set it is not possible to differentiate between different phases in the histogram as it resembles a skewed gaussian. The RC method does, as a result, not provide a good segmentation and detects both too much and too little porosity.

In Paper B the performance of the RC method were evaluated in 2D and 3D for data-sets $\mathcal{A}$ and $\mathcal{B}$ by comparing the detected porosity of each slice. There it was seen that greater care needs to be taken when applying the RC method in 2D as it wrongly classify a portion of the voxels as porous for several slices. For a single slice it also fails as only two different image phases are present in the histogram while three phases are assumed in the computations. Greater accuracy is obtained in 3D as the image phases are better defined in the histogram. For data-set $\mathcal{C}$ the 2D performance of the method was not evaluated due to its low image resolution.

The RC$_m$ approach The RC$_m$ approach includes median filtering of the data prior to computing the globally optimal thresholds using the RC method. The approach was included to illustrate how filtering influences the segmentation result. It is observed that the approach preserves image features, while smoothing image object boundaries and eliminating most of the small pore objects. This is especially true for data-sets $\mathcal{A}$ and $\mathcal{B}$. For data-set $\mathcal{C}$ some image features appears (subjectively) to be better captured. For the last data-set, due to the low resolution, median filtering was applied in a slice-wise manner. The computed porosity is similar to the porosity detected by the RC method, although especially for data-set $\mathcal{A}$ and $\mathcal{B}$ a relatively large number of small pore objects have been removed. Thus, median filtering does not change the main features of the detected pore space.

In Paper B the performance of the RC$_m$ approach was evaluate for data-sets $\mathcal{A}$ and $\mathcal{B}$ in both 2D and 3D, in a similar manner as for the RC method. Similar observations are made for both approaches. Thus, the RC$_m$ approaches is is more sensitive to local grayscale intensity variations in 2D than in 3D.

Otsu's method Otsu's method computes the optimal global thresholds by maximizing the variance between image phases. The computations show that the

method yields nearly identical results as the RC method, only at higher computational cost. The method takes several hours to compute the thresholds, while the RC method is solved within seconds.

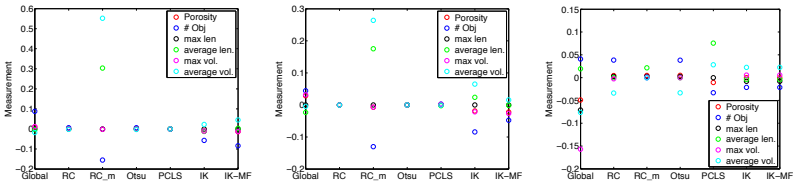
Thresholding by IK Thresholding by IK is a local thresholding method for two-phase image segmentation. The computation time depends on the number of undetermined voxels and it takes from a few days to more than a week, on an external server, to perform the segmentation.

The results show that the method yields a relatively smooth segmentation that limits the number of small regions detected as porosity, which is due to its use of local image information. For both data-sets $\mathcal{A}$ and $\mathcal{B}$ the method is, however, not successful in capturing narrow valleys of porosity. Disappointingly, for these regions the method is only partly successful in separating the pore space from rock matter. For data-set $\mathcal{B}$ some narrow ridges corresponding to rock are also removed. By applying MF prior to IK segmentation an even smoother output is obtained. For data-set $\mathcal{A}$ some narrow channels have been closed and a non-existing channel has been created. For data-set $\mathcal{B}$, MF appears to reinforce some noisy artifacts as image objects are not as well defined.

For data-set $\mathcal{C}$ similar results are obtained both with and without MF prior to IK segmentation. MF was here applied in a slice-wise manner due to the low resolution in the third dimension. It does not have a great effect since the data is already highly blurred. In both cases the results are smooth and image features are to a certain degree captured.

The PCLS method The PCLS method, which is a region based method, is the last method included. The computation show that it yields similar results as the RC method only at higher computational cost. Due to memory issues sub-volume processing is necessary and each sub-volume takes several minutes to compute. The results are later concatenated.

For both data-set $\mathcal{A}$ and $\mathcal{B}$ the method describes the pore space well. As for the RC method the PCLS method struggles to capture narrow valleys of porosity for data-set $\mathcal{B}$. The method does, however, result in overlapping histogram populations that corresponds better to the true histogram populations. This is an advantage of the method and a global thresholding method will never be able to capture this property. Due to regularization the PCLS method results in slightly smoother object boundaries when compared to the RC method, and some smaller pore objects are removed. This is also seen for data-set $\mathcal{C}$. The PCLS segmentation results of data-sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are displayed in Figures 6.1(d)-(f), respectively. For a comparison with the other segmentation methods the reader is referred to Paper B and Paper C.



(a) (b) (c)

Figure 6.2: Normalized pore space measurements subtracted by their median value. The measurements are shown for data-set $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are in (a), (b) and (c), respectively.

The reason the PCLS method produce similar result as the RC method is that they are based on similar ideas formulated in different frameworks. The RC method computes the optimal global thresholds as the midpoint between the means of two adjacent histogram peaks. The PCLS method minimize an energy functional to compute an approximate image $\tilde{I}$ of I_0 and locates the mean phase value $\mathbf{c}$ of each subdomain. By increasing the weight on the regularization term the PCLS method behaves less and less like the RC method, and fewer and smoother image objects are detected. A disadvantage with the PCLS method is that it is more computational expensive and memory demanding when compared to the RC method.

In Paper B the performance of the PCLS method was evaluated in both 2D and 3D for data-sets $\mathcal{A}$ and $\mathcal{B}$. It was shown that the method performs similarly in both dimensions. When compared to the RC method the PCLS method is more computational stable in 2D.

6.2 Pore space characterization

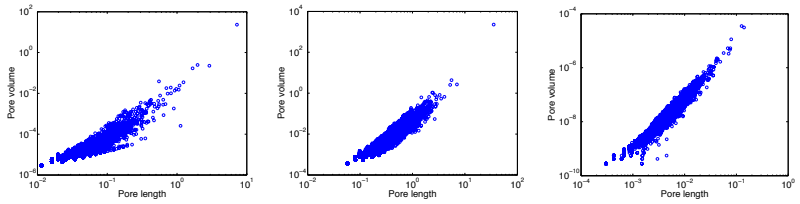
To evaluate the performance of the segmentation methods in 3D various pore space characterizations are computed. The performance is evaluated based on the computed porosity, the computed maximum and average pore object length, and the computed maximum and average pore volume. These measurements are summarized in Table 6.1 for the individual data-sets. In Figure 6.2 they are also visualized after first normalizing the measurements and subtracting their median value. These measurements provided an insight into the pore spaces shape and connectivity. The length of each individual pore object is compute as the maximum *Euclidean* length of the objects convex hull and the pore volume is computed using a simple voxel count.

For data-sets $\mathcal{A}$ and $\mathcal{B}$ it is observed that the global thresholding methods and the PCLS method all detects a large number of image objects. It is generally ob-

Data	Method	Porosity	Image objects				
			# obj.	max len.	average len.	max vol.	average vol.
$\mathcal{A}$	Global	0.2650	40 598	7.2584 mm	0.0137 mm	25.2586 mm^3	0.00064 mm^3
	RC	0.2469	27 776	7.2584 mm	0.0141 mm	23.3998 mm^3	0.00089 mm^3
	RC_m	0.2432	2 479	7.2584 mm	0.0565 mm	23.1733 mm^3	0.00976 mm^3
	Otsu's	0.2468	27 749	7.2584 mm	0.0141 mm	23.3976 mm^3	0.00088 mm^3
	PCLS	0.2468	26 760	7.2584 mm	0.0143 mm	23.4046 mm^3	0.00092 mm^3
	IK	0.2288	17 789	7.2584 mm	0.0124 mm	21.7005 mm^3	0.00128 mm^3
	IK-MF	0.2244	13 643	7.2584 mm	0.0145 mm	21.3235 mm^3	0.00164 mm^3
$\mathcal{B}$	Global	0.2499	32 378	35.4674 mm	0.0921 mm	2.7465 cm^3	0.0872 mm^3
	RC	0.2089	25 836	35.4674 mm	0.1146 mm	2.2550 cm^3	0.0913 mm^3
	RC_m	0.1988	6 460	35.4674 mm	0.2848 mm	2.1480 cm^3	0.3475 mm^3
	Otsu's	0.2088	25 796	35.4674 mm	0.1147 mm	2.2543 cm^3	0.0914 mm^3
	PCLS	0.2108	26 205	35.4674 mm	0.1124 mm	2.2834 cm^3	0.0909 mm^3
	IK	0.1819	13 302	35.4674 mm	0.1378 mm	1.9223 cm^3	0.1545 mm^3
	IK-MF	0.1770	18 717	35.4674 mm	0.1135 mm	1.8476 cm^3	0.1068 mm^3
$\mathcal{C}$	Global	0.0899	7 957	7.6220 cm	0.2589 cm	9.6797 cm^3	0.0187 cm^3
	RC	0.1382	7 848	14.3500 cm	0.2245 cm	79.2686 cm^3	0.0291 cm^3
	RC_m	0.1383	6 122	14.3510 cm	0.2625 cm	80.8171 cm^3	0.0373 cm^3
	Otsu's	0.1386	7 837	14.3500 cm	0.2244 cm	80.3666 cm^3	0.0292 cm^3
	PCLS	0.1245	4 648	14.1006 cm	0.3579 cm	35.1548 cm^3	0.0442 cm^3
	IK	0.1339	5 170	13.3729 cm	0.2177 cm	82.9611 cm^3	0.0428 cm^3
	IK-MF	0.1339	5 172	13.3729 cm	0.2176 cm	82.9615 cm^3	0.0428 cm^3

Table 6.1: Pore space measurements computed from the 3D segmentation results.

served that the RC, Otsu's and PCLS methods yield similar measurements, which is confirmed in Figures 6.2(a)-(b) where it is seen that they all produce measurements that lies approximately at the median value. The other methods deviates from the median value and the RC_m approach has the most variation as it detects a much lower number of pore objects when compared to the other methods. The computed porosity is similar for the RC approaches, Otsu's and PCLS methods, while basic global thresholding based on ϕ_v detects $2-4\%$ more porosity and the IK approaches detects about $2\,\%$ less porosity. Since the detected porosity of the RC_m approach is not reduced too much, especially for data-set $\mathcal{A}$, this indicates that the approach preserves the main image features. This was also observed in the visual evaluation of the methods. The detected porosity of the IK approaches is reduced when compared to the other methods and the number of detected pore objects are reduced as well, disregarding the RC_m approach. This indicates that IK segmentation does not preserve image features to the same degree as in the other methods. To a certain degree similar observations can be made from the computed maximum pore object, which is smaller than for the other methods. This was also observed in the visual evaluations. For all methods the maximum pore objects length runs approximately along the image volumes diagonal from



(a) (b) (c)

Figure 6.3: Pore length versus pore volume plots after PCLS segmentation on data-set $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are shown in (a), (b) and (c), respectively.

one side to the other, indicating that the pore space is interconnected throughout the rock sample. Since the average pore object length and volume are small, the majority of pore objects are small as well. This is also shown in Figure 6.3, which displays scatterplots of the pore length versus pore volume measurements. There it is observed a single larger pore object has been detected for data-sets $\mathcal{A}$ and $\mathcal{B}$ after PCLS segmentation, while the remaining pore objects are small in size.

For data-set $\mathcal{C}$ there are more variations in the segmentation results which is reflected in more variations in the computed pore space measurements as well. This is confirmed by Figure 6.2(c), there is generally more more variation when compared to the two other data-sets. One exception is the RC_m approach whose measurements lies approximately at the median value. The most deviation is seen for basic global thresholding based on ϕ_v, which is likely due to underestimation of the pore space. This approach is therefore not considered further her. The other methods detect similar amount of porosity, with the exception of the PCLS method. It detects less porosity which is probably because it detects the smallest number of pore objects as well. The computed maximum volume is similar for the methods, with the exception of the PCLS method. There the maximum volume is much smaller, which is probably due to splitting the maximum volume detected by the other methods into parts. This can also be observed in the scatterplots in Figure 6.3(c), where it is observed that two major pore objects are detected that are similar in length and volume. In addition there exists several other pore objects that are larger in size, while the remaining objects are small. The IK approaches detects smoother image objects, which is probably the reason why they also detects shorter maximum object lengths, but larger maximum volume when compared to the other approaches. For all methods, the computed maximum pore objects length does not run through the image volume and only the major vugs are detected with a fair amount of accuracy.

6.3 Conclusion

The objective of this work was to test the performance of various image segmentation techniques for segmentation of CT images of vuggy carbonate rock. Three different 3D image data-sets are encounter. Two of them are acquired using a μCT scanner, while the third one is acquired using a CT scanner. The image data-sets have varying degrees of difficulty since they have varying image resolution. The computations show that the methods generally perform well on high resolution μCT scans. For decreasing image resolution the accuracy of the segmentation also decrease.

For the two μCT scans it is observed that the RC, Otsu's and PCLS methods all provide similar segmentation results. The RC is the fastest to compute, while Otsu's method take significantly longer time and for the PCLS method sub-volume processing is necessary due to memory issues. The PCLS method does, however, detect overlapping histogram populations, which better represents the overlapping histogram populations of the input image. Filtering prior to RC thresholding yields smoother object boundaries and reduces the number of small pore objects, while preserving the main features. This illustrates that if the general trend is of interest filtering prior to segmentation is a good option. The remaining approaches does not perform satisfactory. Global thresholding based on ϕ_v detects too much porosity and it does not take into account the available image histogram nor image information. The IK approaches have a tendency to wrongly classify narrow valleys of porosity as rock. This was disappointing as great expectations were held to the methods performance. MF prior to IK segmentation yields worse results.

For the third low resolution CT scan the included methods yields similar results. As this data-set is highly blurred it is challenging to segment and also challenging to evaluate the methods performance. It is observed that the methods are only able to detect the major vugs, but not the fine details. Global method will never provide a good segmentation for this data-set as its histogram has the shape of a skewed gaussian. Therefore they both over- and underestimates the pore space. This is a problem for the PCLS and IK methods as well, and some other method might be more appropriate. Further, since the data-set does not have well defined image objects it might also only be realistic to detect major vugs.

Pore space measurements show that the pore space of the two high resolution μCT scans is interconnected throughout the image volume. There exists a single large pore object while the remaining objects are small in size. This is also confirmed by scatterplots and corresponds well with the knowledge that the pore space is interconnected at μ-scale. The detected porosity is less than the expected visible porosity, ϕ_v, which leaves room for some improvements.

The third data-set is not shown to be interconnected throughout the pore space.

Since this sample is acquired from the same block of rock as the two other samples it is reasonable to assume that this sample is interconnected throughout the pore space at μ-scale as well. This porosity is, however, located below image resolution. For this data-set ϕ_v was most likely underestimated, which explains why the methods detest more porosity.

For both μCT scans the RC, Otsu's and PCLS methods yield similar results. Both RC and Otus's method computes similar globally optimal thresholds, although the computation time increase significantly for Otsu's method when compared to the RC method. The PCLS method detects overlapping histogram populations, which better represents the overlapping populations in the image histogram when compared to the iterative global thresholding methods. It was also shown to be more stable for 2D segmentation than the RC method. The method is, however, more time consuming to solve than the RC method. Due to memory issues sub-volume processing is also required for solving the PCLS method. Since the results of the methods are similar, this indicates that the RC method can be used for initial investigations. For a more accurate segmentation the PCLS method is a more appropriate choice. Furthermore, by increasing the weight on the regularization term the PCLS method will behave less and less like the global methods, and fewer and smoother objects are detected.

6.4 Future work

It would be interesting to investigate additional segmentation methods for segmentation of the vuggy carbonate rock image data-sets. For instance, Markov random fields might be investigated. It makes makes use of statistical information and can be solved by using graph cut as was the case of the PCLS method. Another method that could be of interest is the local multi-phase thresholding method in [17] that combines statistical computations with the image gradient to complete the segmentation. Furthermore, since the NMR measurements yields information about the pore size distributions, it would be interesting to investigate if this knowledge could be included in a statistical model to guide the segmentation at the different scales.

In the recent years graphical processing unit (GPU) implementations are becoming increasingly popular for faster computations. It would be interesting to implement the segmentation methods on the GPU to develop faster solvers.

In the future it would also be interesting to look into pore network modeling. Following the work of [27, 9] it would be interesting to investigate the maximum ball approach in which spheres are grown for each voxel in the pore space. The larger spheres are regarded as pore objects while chains of smaller spheres are viewed as pore throats. Single- and two-phase fluid flow simulation on this model

would also be of interest. A disadvantage of modeling is that models mimics the real pore space without necessarily capturing the pore spaces complexity.

Another approach is to perform computations directly on the binary pore space. This approach has been used by e.g. [33], where they modeled single- and two-phase fluid flow using the Lattice Boltzmann method. The approach is computationally expensive. To reduce the computation time the computations were performed directly on the GPU.

It would also be interesting investigate fluid flow modeling and simulations in which properties computed at different scales are taken into account. These properties can be found either through laboratory experiments or be computed from segmentations at different scales.

Once fluid flow models have been constructed and fluid flow simulations have been performed, it would be interesting to compare the results with laboratory fluid flow experiments. Ideally, realistic simulations should give similar results as the laboratory experiments.

Appendices

Appendix A

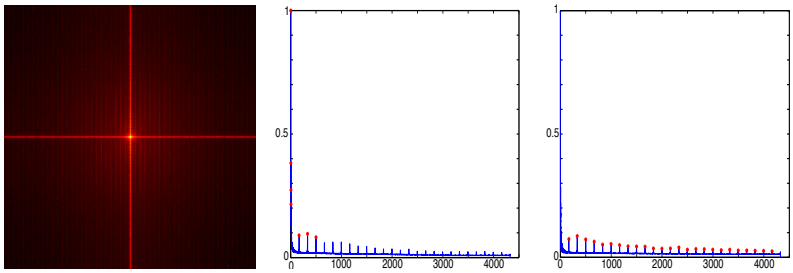
Determine notch filter intervals

After evaluating frequency spectrums of input images, such as the one displayed in Figure A.1(a), it is clear that there are multiple vertical frequency ridges corresponding to image artifacts. To precisely attenuate these ridges it is necessary to precisely located their center point. This was facilitated by an observation that a 1D signal, $C(u)$, can be computed by summing over each frequency domain column, which yields a signal with peaks at the ridge center points. It was computed by

$$C(u) = \sum_{v=0}^{N-1} |F(u+a, v)|, \quad \text{for } u = 0, \ldots, \frac{M}{2} - 1\,,\ a = \frac{M}{2}\,, \tag{A.1}$$

and then normalized to ensure that it only has values in range $[0,\ 1]$. Due to symmetry properties of the spectrum it is sufficient to compute $C(u)$ for the spectrums positive half. An example of a typical 1D signal is shown in Figure A.1(b).

Locating the peaks of $C(u)$ is then performed by a two-step procedure. In the *first* step the matlab program *peakdet.m*, developed by [3], is applied to locate the first peak. It takes one parameter Δ and locates the local maximums of a 1D signal. A point is considered a local maximum if it has a maximum value, and was preceded (to the left) by a value lower than Δ [3]. Choosing Δ too small will result in too many detected maximums as the computed 1D signal can be noisy. Using this program we are only able to locate a few true peaks as is shown in Figure A.1(b), illustrated by the red dots. Some falsely detected peaks are obtained as well. Observe that all true peaks are located with approximately the same distance to the neighboring peaks. In the *second* step, using this observation, the remaining peaks are located using a search algorithm where the first true peak is used as a starting point. It works by first discarding the falsely detected peaks. The spacing between the origin and the first true peak is then used to search for the second peak. The maximum peak within a predefined neighborhood is then



(a) (b) (c)

Figure A.1: In (a) a typical frequency domain spectrum is shown and in (b) the computed 1D signal $C(u)$ of the spectrums positive half. The red dots are the detected peaks by *peakdet.m* [3]. In (c) the falsely detected peaks have been discarded and the remaining peaks have been found using a simple search algorithm.

selected as the center point of the second ridge. The remaining peaks are found in a similar manner. In Figure A.1(c) it is observed that all peaks are accurately detected by the procedure.

Acronyms

1D one-dimensional

2D two-dimensional

3D three-dimensional

CT computed tomography

DFT discrete Fourier transform

GPU graphical processing unit

IK indicator kriging

IDFT inverse discrete Fourier transform

MF majority filtering

NMR nuclear magnetic resonance

OK ordinary kriging

PCLS piecewise constant level set

RC Ridler and Calvard's

TV total variation

Bibliography

[1] C.H. Arns, F. Bauget, A. Limaye, A. Sakellariou, T.J. Senden, A.P. Sheppard, R.M. Sok, W.V. Pinczewski, S. Bakke, L.I. Berge, , P.-E. Øren, and M.A. Knackstedt. Pore-scale characterization of carbonates using x-ray microtomography. *SPE Journal*, 10(4):475–484, December 2005.

[2] Egil Bae and Xue-Cheng Tai. Graph cuts for the multiphase Mumford-Shah model using piecewise constant level set methods. *UCLA Computational and Applied Mathematics Reports (08-36)*, 2008.

[3] Eli Billauer. Peak detection using matlab. Matlab program, September 2011. Downloaded August 29, 2011.

[4] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transaction on*, 26(9):1124–1137, 2004.

[5] Tony F. Chan and Luminita A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.

[6] Noel A.C. Cressie. *Statistics for Spatial Data*. Wiley series in probability and mathematical statistics. John Wiley & Sons, Inc., revised edition, 1993.

[7] Clayton V. Deutsch. Correcting for negative weights in ordinary kriging. *Computers & Geosciences*, 22(7):765–773, 1996.

[8] Ketil Djurhuus, Bartek Florczyk Vik, and Arne Skauge. Pore size characterisation of vuggy carbonate material. 2008 International Symposium of the Society of Core Analysts (SCA), Abu Dhabi, October 2008.

[9] Hu Dong and Martin J. Blunt. Pore-network extraction from micro-computerized-tomography images. *PHYSICAL REVIEW E*, 80, 2009.

[10] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[11] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Pearson Education, Inc., 3 edition, 2008.

[12] Edward H. Isaaks and R. Mohan Srivastava. *An Introduction to Applied Geostatistics*. Oxford University Press, 1989.

[13] Andre G. Journel. Fundamentals of geostatistics in five lessons. American Geophysical Union, 1989. Short Course in Geology: Volume 8.

[14] Richard A. Ketcham and William D. Carlson. Acquisition, optimization and interpretation of x-ray computed tomographic imagery: applications to the geosciences. *Computers & Geosciences*, 27(4):381–400, 2001.

[15] Johan Lie, Marius Lysaker, and Xue-Cheng Tai. A variant of the level set method and applications to image segmentation. *Mathematics of Computation*, 75(255):1155–1174, 2006.

[16] W. Brent Lindquist and Arun Venkatarangan. Pore and throat size distributions measured from synchrotron x-ray tomographic images of fontainebleau sandstones. *Journal of geophysical research*, 105(B9):21,509–21,527, September 2000.

[17] Yinxiao Liu, Guoyuan Liand, and Punman K. Saha. A new multi-object image thresholding method based on correlation between object class uncertainty and intensity gradient. *Medical Physics*, 39:514–532, 2012.

[18] F. Jerry Lucia. *Carbonate Reservoir Characterization*. Springer-Verlag Berlin Heidelberg, 1999.

[19] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[20] Wonho Oh and W. Brent Lindquist. Image thresholding by indicator kriging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(7):590–602, 1999.

[21] Stanley Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer-Verlag New York, Inc., 2003.

[22] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[23] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[24] John G. Proakis and Dimitris G. Manolakis. *Digital signal processing: Principles, Algorithms, and Applications*. Prentice-Hall, Inc., 3 edition, 1996.

[25] T. W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8(8):630–632, 1978.

[26] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[27] Dmitriy Silin and Tad Patzek. Pore space morphology analysis using maximal inscribed spheres. *Physica A*, 371:336–360, 2006.

[28] A. Skauge, T. Horgen, B. Noremark, and B. Vik. Experimental studies of unstable displacement in carbonate and sandstone material. In *16th European symposium on improved oil recovery*. European association of geoscientists & engineers (EAGE), April 2011.

[29] A. Skauge, K. Sorbie, P.A. Ormehaug, and T. Skauge. Experimental and numerical modeling studies of viscous unstable displacement. In *15th European Symposium on Improved Oil Recovery*. European association of geoscientists & engineers (EAGE), April 2009.

[30] Arne Skauge, Per Arne Ormehaug, Tiril Gurholt, Bartek Vik, Igor Bondino, and Gerald Hamon. 2-d visualisation of unstable waterflood and polymer flood for displacement of heavy oil. In *Eighteen SPE improved oil recovery symposium*. Society of petroleum engineers (SPE), April 2012.

[31] S. R. Stock. X-ray microtomography of materials. *International Materials Reviews*, 44(4):141–164, 1999.

[32] Xue-Cheng Tai, Oddvar Christiansen, Ping Lin, and Inge Skjælaaen. Image segmentation using some piecewise constant level set methods with MBO type of projection. *International Journal of Computer Vision*, 73(1):61–76, 2007.

[33] Jonas Tolke, Chuck Baldwin, Yaoming Mu, Naum Derzhi, Avrami Grader, and Jack Dvorkin. Computer simulations of fluid flow in dediment: From images to permeability. *The Leading Edge*, pages 68–74, January 2010.

[34] Luminita A. Vese and Tony F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah mode. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[35] B. Vik, K. Djurhuus, K. Spildo, and A. Skauge. Characterisation of vuggy carbonates. *SPE/EAGE Reservoir Characterization and Simulation Conference, Abu Dhabi*, 2007.

[36] J. Wang and M. Dong. Optimum effective viscosity of polymer solution for improving heavy oil recovery. *Journal of Petroleum Science and Engineering*, 67(3-4):155–158, August 2009.

[37] D. Wildenschild, J. W. Hopmans, C. M. P. Vaz, M. L. Rivers, D. Rikard, and B. S. B. Christensen. Using x-ray computed tomography in hydrology: systems, resolutions, and limitations. *Journal of Hydrology*, 267:285–297, 2002.

Part III

Included Papers

Summary of the papers

This thesis include four scientific papers. They cover different topics, which is a reflection of the various topics considered in this work. In the following a short summary of the papers is given.

Paper A: 2-D Visualisation of Unstable Waterflood and Polymer Flood for Displacement of Heavy Oil. In the proceedings of the Eighteenth SPE Improved Oil Recovery Symposium, Tulsa, Oklahoma, USA, 14-18 April 2012.

Arne Skauge, Per Arne Ormehaug, Tiril Gurholt, Bartek Vik, Igor Bondino and Gerald Hamon

The objective of this paper was to obtain improved knowledge of oil mobilization processes through imaging of fluid flow experiments. Imaging was performed using an in-house X-ray scanner which introduces horizontal and vertical line artifacts. These artifacts are either periodic or aperiodic, and have varying intensities. Image filtering was applied to improve the presentation of the image data. Due to the periodic behavior of parts of the artifacts it was natural to consider frequency domain filtering. Good results are obtained and the line artifacts are largely attenuated without distorting the fluid flow. The author of this thesis contributed by construction of the frequency domain filtering routine as well as performing the filtering and scientific discussions. The author also contributed in writing parts of the article.

Paper B: Determination of Connectivity in Vuggy Carbonate Rock Using Image Segmentation Techniques. In the proceedings of the 12^{th} European Conference on the Mathematics of Oil Recovery, ECMOR XII, Oxford, UK, 6-9 September 2010.

Tiril Gurholt, Bartek Vik, Ivar Aavatsmark and Sigurd Aanonsen

The objective of this paper was to evaluate the performance of image segmentation techniques applied for segmenting CT imaged vuggy carbonate rocks. Their performance were evaluated based on the segmentation of three different data-sets with varying image resolutions. It was shown that the methods perform well on high resolution μCT scans. As the resolution decrease the accuracy in the segmentation also decrease. Similar results were obtained in both 2D and 3D, although in 2D there is a greater risk of wrongly classifying parts of voxels for some of the approaches. The author of this thesis contributed by planning and performing the experiments, implementation of the segmentation methods, analysis of the results and scientific discussions, as well as writing the article.

Paper C: Pore space characterization of vuggy carbonate rocks: A comparative study of the performance of various image segmentation techniques. Submitted to Journal of Petroleum Science and Engineering, June 2013.

Tiril Gurholt and Bartek Vik

The objective of this paper was to evaluate the performance of various image segmentation techniques applied for segmenting CT imaged vuggy carbonate rocks. This paper is an extension of Paper B in which an additional couple of segmentation methods have been included. Similar results are obtained and the additional segmentation methods do not provide an improvement. It was observed that the PCLS method yields similar results as iterative global threshold methods at higher computational cost when applied μCT scans. Since the method detects overlapping histogram population the segmentation is more accurate than the global methods. For a low-resolution CT scan there are more variation in the results and it is challenging to evaluate the methods performance. The author of this thesis contributed by planning and performing the experiments, implementation of the segmentation methods, analysis of the results and scientific discussions, as well as writing the article.

Paper D: 3D Multiphase Piecewise Constant Level Set Method Based on Graph Cut Minimization. Published in Numerical Mathematics: Theory, Methods and Applications, Volume 2, Number 4, Pages 403-420, November 2009.

Tiril Gurholt and Xuecheng Tai

The objective of this paper was to minimize the PCLS method by graph cut in 3D. The PCLS method is a region based segmentation approach for multiphase problems. It minimizes an energy functional to approximate an image by a piecewise constant image function. This is traditionally done by deriving and solving an Euler-Lagrange equation. In this work it was shown that the method can be minimized by graph cut in 3D by following the ideas of an existing 2D solver. To solve the segmentation problem the minimum cut on a special graph needs to be computed. The approach yields a fast and accurate solver, and the number of phases can be overdetermined. Using the graph cut solver the PCLS method is solved faster than the traditional Euler-Lagrange approach. The author of this thesis contributed by planning and performing the experiments, developing the 3D solver in C++, scientific discussions and writing the article.