Measurements on microtomographic images of fibrous structures

Jens Bache-Wiig  
bachewii@stud.ntnu.no
Per Christian Henden  
perchrh@pvv.org

November 2004

Project report in TDT4725
Image Processing specialization
Supervised by Professor Richard Blake

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Computer and Information Science

Version 1.03
Preface

This is the report on a study conducted by two computer science major students at the Department of Computer and Information Science at the Norwegian University of Science and Technology. The study is part of preparatory work for the master thesis. The project’s period was from the 16th of August to the 26th of November. The project assignment was specified by the Paper and Fibre Research Institute.

The project was supervised by Prof. Richard Blake from the Norwegian University of Science and Technology (NTNU), and Dr. Ing. Per Nygård from the Paper and Fibre Research Institute (PFI).

Structure of the report

The first chapter gives an introduction to the report and the main methods of image acquisition in the paper-science domain. Chapter two explains central concepts in computer science and looks at how to make the processing of high-resolution, three-dimensional images more feasible. The third chapter gives an overview of the existing method of processing images of material samples. Improvements are suggested. Chapter four gives an overview of measures done on these images in the paper science domain. The final chapter introduces fractal analysis and looks at how it can be used to make measurements on images.

We specify which sections are our own work throughout the report. Each chapter generally has two parts: a literature study and our own work, and in that order.

Intended audience and how to read this report

It has been assumed that the reader has a basic knowledge of computer science and mathematics. Concepts in paper science, physics, image analysis and graphics are explained where the understanding of these concepts are necessary to understand other parts of the report. Concepts that are less central are not explained, but the reader is given references that enables her to learn these concepts from other sources. This explanation of common concepts is necessary for the report to be read by people from all fields interested in the analysis of high-resolution microscopy images, which includes material science, biology, chemistry, computer science and others. We have included an index at the end of this document so that the reader may get back to definitions and explanations of concepts if necessary.
Acknowledgements

The authors would like to thank Richard Blake, Per Nygård, Øyvind Gregersen and Gary Chinga for their suggestions and feedback on the project. A special thanks goes to Anneli Olsbø for proof-reading a draft of this report.
Abstract

Recent improvements in computer processing and storage capabilities have made the analysis and visualisation of high-resolution three-dimensional images of material samples possible. The structure of the material is of special significance to its properties, and has as such been an active area of research. In order to analyse the structure, noise-reduction and binarisation of the material images are necessary. We have examined the current methods of noise-reduction and binarisation and suggested improvements to these. The different methods of quantification of material properties from their images in the literature has been summarised and a new method of structure quantification based on fractal analysis is suggested.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>iv</td>
</tr>
<tr>
<td>Abstract</td>
<td>v</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objective</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Image acquisition</td>
<td>2</td>
</tr>
<tr>
<td>2 Volume graphics</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Volumetric data</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Visualisation</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Reducing storage demand</td>
<td>12</td>
</tr>
<tr>
<td>2.4 Improving performance</td>
<td>16</td>
</tr>
<tr>
<td>2.5 Conclusions</td>
<td>18</td>
</tr>
<tr>
<td>3 Binarisation</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>19</td>
</tr>
<tr>
<td>3.2 Current method of binarisation</td>
<td>19</td>
</tr>
<tr>
<td>3.3 Discussion of current method</td>
<td>22</td>
</tr>
<tr>
<td>3.4 Suggested improvements</td>
<td>23</td>
</tr>
<tr>
<td>3.5 Preliminary results and discussion</td>
<td>26</td>
</tr>
<tr>
<td>4 Feature extraction</td>
<td>33</td>
</tr>
<tr>
<td>4.1 Existing methods</td>
<td>33</td>
</tr>
<tr>
<td>4.2 New methods</td>
<td>43</td>
</tr>
<tr>
<td>4.3 A practical application of feature extraction</td>
<td>43</td>
</tr>
<tr>
<td>4.4 Relevance to new materials</td>
<td>44</td>
</tr>
<tr>
<td>4.5 Discussion and future work</td>
<td>45</td>
</tr>
<tr>
<td>5 Fractal analysis</td>
<td>47</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>47</td>
</tr>
<tr>
<td>5.2 Mathematical foundation</td>
<td>48</td>
</tr>
<tr>
<td>5.3 Fractal dimension as a descriptor</td>
<td>49</td>
</tr>
</tbody>
</table>
## Contents

5.4 Estimating the fractal dimension .......................... 50  
5.5 Fractal dimension and roughness .......................... 55  
5.6 Discussion and future work ................................. 58  

6 Conclusion .................................................. 59  

A Implementation ............................................. 61  
  A.1 SUSAN plugins ........................................... 61  
  A.2 Plugins for fractal analysis ............................... 62  

Bibliography .................................................... 65
Chapter 1

Introduction

This chapter establishes the background for our report. The two main methods for high-resolution, three-dimensional image acquiring are presented and compared.

1.1 Background

In material science it is desirable to learn more about how material properties at the micro level influence the macroscopic level (Holmstad et al. 2003). Manual or automated image analysis of three-dimensional images of material samples is a useful tool in this respect. Automated analysis has the potential to be more exact, reliable and objective than subjective analysis.

Fibre structures are generally highly interwoven, porous three-dimensional structures. To give a complete analysis of such structures it is reasonable that we should base our measurements on fully three-dimensional data. Only recently has the technology for acquiring and processing large image volumes made such studies practical. Yang (2001) advocates that the third dimension is an important factor in dealing with thick fibrous materials, and that using fully three-dimensional volume images will bring major improvements in the analysis of fibre structures.

Several previous projects at PFI, the Norwegian Paper and Fibre Research Institute, have focused on analysing microscopic images of paper. Some of the most recent contributions include (Holmstad 2004, Arns et al. 2004, Wang et al. 2004, Holen and Hagen 2004). In order to compare different material samples, a quantification of material properties based on the images of the material samples is calculated. PFI is planning to extend these efforts to other materials, such as cellulose-composites and fibre-based absorbent materials. This calls for a summary and evaluation of existing techniques, as
a thorough understanding of existing techniques is assumed to better enable the adaption of existing methods and development of new methods for use in the analysis of new materials.

1.2 Objective

The project aims to give an overview of the current state of feature extraction from three-dimensional tomographic images of paper, including the process of preparing the images for feature extraction. If possible, these areas should be extended and improved upon. The feature fractal dimension is such an extension of special interest in that regard.

A study of techniques for reducing the resource demands associated with the analysis of high-resolution three-dimensional images should be performed in order to make such analyses more feasible.

The report should aim to be comprehensible for paper scientists.

1.3 Image acquisition

The quality of an image depends on the method by which it is acquired. The way our image data is acquired, is perhaps the most unique aspect of the research presented in this report. Because a basic understanding of the acquisition is assumed in many of the following chapters, we will present it in this section.

There are several methods for acquiring high resolution three-dimensional image data. Some common examples are Confocal laser scanning microscopy (CLSM), scanning electronic microscopy (SEM) or light microscopy (LM) combined with serial sectioning, magnetic resonance imaging (MRI), and X-ray micro tomography. According to Holmstad (2004, p. 86), the preferred technique for acquiring three-dimensional is X-ray micro tomography because “it gives access to the paper structure non-invasively at a uniform high resolution and contrast in all spatial directions and all parts of the imaged paper volume”.

Unlike SEM- or LM-based techniques based on serial sectioning, one does not have to cut into the material in order to obtain image slices. This avoids distortion between image slices, and simplifies the reconstruction of the digital volume.

The other applicable methods have either varying contrast or resolution in the different spatial directions. For instance, all invasive techniques require
precise cutting of cross sectional slices through the material. Even though such methods often give better quality images and a higher resolution for each slice, it is difficult to achieve good resolution along the direction of the sectioning. Hence, the resulting voxels are oblong, complicating the data analysis.

The main disadvantages of X-ray tomography is its currently high cost and low availability. Desktop-sized equipment exists, but supplies lower resolution images than currently available large synchrotron sources. According to Holmstad (2004), the current achievable resolution is roughly $5\mu m$, but future developments will likely make high resolution X-ray tomography more affordable and available.

1.3.1 X-ray micro tomography

The creation of three-dimensional tomographic volumes with resolution considerably below the centimetre scale is commonly referred to as micro tomography (X-\(\mu\)CT). Resolution below $1\mu m$ require high energy photons with good penetration abilities. The X-ray beam also requires a high degree of precision. These properties are currently only achievable through the use of a large, circular particle accelerator known as a synchrotron (Holmstad 2004, p. 86).

![Figure 1.1: The basic principle behind X-ray tomography. Figure reproduced from http://www.esrf.fr](http://www.esrf.fr)

Figure 1.1 shows the basic working principle of X-ray tomography. A series of projected images are formed from the X-rays passing through the sample, while the sample is being rotated $180^\circ$. This results in a projected radiograph for every angle. The set of radiographs are used to reconstruct the three-dimensional structure by solving a Fourier-like integral, known as the Radon transform (Holmstad 2004, p. 81).
Sample preparation

Samples are prepared by cutting out a small rectangular material sample. The sample is then mounted on a thin capillary by using melted glue. The capillary and the glue are highly visible features in the resulting data (figure 1.2).

![Figure 1.2: Typical absorption mode image showing the fibre structure, capillary and glue.](image)

There are two different methods that can currently be used to generate the projection images. These methods are complimentary, in that they have different properties, and can be used for different purposes. The two methods are presented in the following sections. The images available to us are obtained at the European Synchrotron Radiation Facility (ESRF), beamlines ID-22 and ID-19. The size of the sampled volume is roughly $1\text{mm}^3$. The exact dimensions will depend on the particular material sample.

Absorption mode imaging

Different contrast in absorption mode images reflects different levels of beam-absorption in the material. The absorption level is generally determined by the average atomic density of the material. Figure 1.3 illustrates this procedure. This process is analogous to the X-ray images normally used in hospitals, where bone structure results in a higher intensity than softer tissue. Unfortunately some organic materials studied at PFI do not have enough absorption contrast to make this method efficient. The resolution obtainable by this technique at ESRF is approximately $0.7\mu\text{m}$.

Phase contrast imaging

Phase contrast imaging is illustrated in figure 1.4. In phase-contrast imaging the amplitude of the signal is, according to constructive and destructive
interference of the X-ray waves, refracted at phase borders in the sample (Holmstad 2004, p. 95), in other words high contrast in the resulting image represents a border between regions of a different refractive index. The resolution obtainable by this technique at ESRF laboratories, is approximately 0.35µm.

![Figure 1.3](http://www.esrf.fr)

**Figure 1.3:** Illustration of absorption mode imaging. Figure reproduced from [http://www.esrf.fr](http://www.esrf.fr)

Comparison and discussion

Image data generated by near-field phase-contrast, which contain only region boundaries, are generally more difficult to binarise than data generated by absorption mode. To binarise such volumes, regions must be filled. This filling-procedure is complicated by the region boundaries being broken up by noise and other processes. Absorption mode imaging results in complete regions and does not require region-filling.

The advantage of using phase-contrast imaging to absorption mode is that it has a higher spatial resolution. For some lighter materials, phase-contrast is still the only feasible method available.
The binarisation of phase contrast images is thoroughly treated in Antoine et al. (2001). The article gives a 23-step procedure for manual binarisation of such images. The key points in this procedure are three-dimensional smoothing by averaging, low-pass Butterworth filtering in the frequency domain, median filtering and local thresholding (a thorough treatment of these concepts is given by Sonka et al. (1999)). In the mentioned procedure, seeding points has to be manually selected for each fibre of interest in every ninth image slice, in order to fill the regions in the volume. Another drawback is the extensive use of low-pass filtering, which significantly reduces image details.

Prior research in the field has mostly concentrated on the processing of phase contrast data, as it has been assumed (Antoine et al. 2001, Ramaswamy et al. 2001) that X-ray imaging of organic fibrous materials like paper requires the use of phase-contrast imaging. Recent advances at ESRF, beam line ID-19, resulted in absorption mode images with a better contrast than previously assumed achievable. Using absorption mode is the preferred method due to the simpler processing. Although phase-contrast imaging can provide higher spatial resolution than absorption mode, the effective resolution is degraded by the amount of filtering necessary to binarise the data in the images. Recent research (Goel et al. 2004, Holmstad 2004), have concluded that absorption mode X-ray micro tomography is well suited for the determination of quantitative measures of the paper structure.

Holmstad (2004, p. 138) discourages further development of image processing of near-field phase-contrast three-dimensional images. Because of the promising results achieved using absorption mode imaging and Holmstad’s recommendation, we will focus on the processing of absorption mode images in the rest of this report.
Chapter 2

Volume graphics

This chapter aims to give a brief overview of how to handle volumetric image data in a resource-saving manner. We will start by defining some general concepts relating to volume graphics. We will also give an overview of how to visualise such data and investigate general methods on how to efficiently store and process them.

2.1 Volumetric data

The main division in computer graphics goes between discrete and continuous representations. Three-dimensional computer graphics are often based on continuous representations, but to measure quantities in the real world, we have to make discrete measures. Digital photography is a typical example, where the light is measured only at discrete intervals across a large matrix.

Volumetric data is typically a set $S$ of samples $(x, y, z, w)$, representing the value $w$ of some property at a specific location $(x, y, z)$. If the range of values is restricted to either 0 or 1, we refer to such data as binary data. Alternatively the value may represent some measurable property of the data, including grey-level or atomic density at each location. In the case of multispectral data, e.g. colour images, the data is described by multiple sets.

In general such samples may be taken at random locations in space, but in most cases the set $S$ consists of samples taken at regularly spaced intervals along three orthogonal axes. When the interval on each axis is of equal length, the set $S$ is called isotropic, otherwise it is called anisotropic (Kaufman 1994). It is most common to store such structures in a three-dimensional array, with the element location used to indicate its position on the grid. The set $S$ will be referred to as the array of values $S(x, y, z)$,
defined only on grid-locations. If we consider the grid-locations to be centre-points on equally spaced cuboids\footnote{Also known as a rectangular parallelepipeds} or cubes, the individual cubes are often referred to as voxels. Digitised image elements (the two-dimensional case) are commonly referred to as pixels.

### 2.1.1 Sampling and transformations

Since the set $S$ is only valid at discrete intervals, a function $ip(x,y,z)$ that gives an estimate of the value at any continuous location may be needed. $ip$ must be defined so that $ip = S$ on grid locations, but on non-grid locations $ip$ returns the value interpolated from nearby grid locations. The simplest such function is the nearest neighbour method, also called zero order interpolation, that simply returns the value of the voxel containing it. First or higher order interpolations can be used to achieve better approximations.

![Figure 2.1: Demonstration of the problems associated with voxel transformations](image)

A considerable problem with discrete representations is that they are notoriously difficult to transform. This includes our pixels and voxels. A thorough treatment of pixel coordinate transformations can be found in Sonka et al. (1999). This explanation is easily extended to voxels as well. Any transformation applied requires that data is fitted onto another discrete voxel-grid, and this is usually not a simple one-to-one mapping. For each grid point in the target grid, a sample is taken at the transformed location in the original grid. Because this transformed point tends to not be exactly centred on a voxel, an interpolation is necessary, and can be performed by using the function $ip(x,y,z)$ defined in the previous paragraph.

Figure 2.1 demonstrates how discrete data is affected by rotating a line
2.2. Visualisation

35° clockwise and back again using voxel transformation. The upper row shows the transformation performed using zero order interpolation, while the second row shows the result from a first order interpolation routine. Much information about the original voxel configuration is lost in the process (figure 2.1).

![Figure 2.2: 26-adjacent (1), 18-adjacent (2) and 6-adjacent (3) voxels](image)

2.1.2 Adjacency and connectedness

According to Kaufman (1994), two voxels are said to be 26-adjacent if they share either a corner, an edge or a face. Voxels that share either an edge or a face are said to be 18-adjacent and those sharing only a face are said to be 6-adjacent. These N-adjacencies are illustrated in figure 2.2. The equivalent adjacencies for two-dimensional pixel-grids are the 4-adjacent and 8-adjacent pixels, sharing either edges or both edges and corners respectively.

We define the prefix $N$ to define a particular adjacency relation. A sequence of voxels that share a common property where all consecutive voxels are $N$-adjacent, is an $N$-path. A set $W$ of voxels is called $N$-connected if there is an $N$-path between every pair of voxels in $W$.

2.2 Visualisation

In order to discover new correlations between physical properties from three-dimensional images of the material, human experts may benefit from an accurate visual representation of the image data, as this can improve the expert’s understanding of the material structure. The simplest form of representation is to present consecutive image slices while scrolling through the voxel volume in different directions. This will only yield a limited three-dimensional representation, and is not dependent on defining a surface because we are in effect looking inside the material.

Several other forms of visualisation have been presented in Kaufman (1994). These are strictly divided into surface rendering techniques and volume ren-


2.2.1 Surface rendering techniques

This technique requires the explicit definition of a surface. A defined surface requires that the data have been successfully binarised, as described in Chapter 3 and that a procedure known as surface fitting is applied. Surface fitting consists of mapping geometric primitives onto the binarised data so that a surface is created between objects and void. Because the surface is a continuous structure, we must use some principle of interpolation between the discrete sample grid.

When zero order interpolation is used on the voxels, squares are mapped onto the voxels wherever a boundary between object and background is detected. Such simple geometrical primitives are often referred to as polygons. A depth order algorithm is used to draw shaded squares onto the image plane. The result is a blocky, but comprehensible, image. This blocky image is unsuitable for assessment of optical properties, because light beams hitting voxels will be reflected only at $90^\circ$ angles.

Surface mappings of higher quality can be achieved by defining the surface using higher order interpolation. Such surfaces are known as iso-valued surfaces. For accurate simulation of optical properties, it is necessary that good approximations of an iso-valued surface is made. One of the most popular algorithms is the Marching Cubes algorithm, introduced by Lorensen and Cline (1987). It approximates iso-surfaces by limiting the ways a surface can pass through a voxel to one of 256 different predetermined polygon configurations, including triangular polygons. By symmetry this number can be reduced to 15 unique configurations, each stored in a look-up table. Each voxel is then mapped onto one of these configurations. By calculating the face normals for each triangle, and then generating averaged normals in each corner of each polygon with its neighbours, a smooth looking surface can be rendered.

Holmstad (2004) concludes that the great amount of surface data generated by the marching cube algorithm prohibited the inspection of the entire image volume acquired through synchrotron X-ray tomography, on current commodity hardware. However, larger memory capacity and rapidly improving surface graphics acceleration, will probably make real-time inspections of iso-surface data from tomographic images a reality.

The improvement of iso-surface mappings is an area of active research.
2.2. Visualisation

2.2.2 Volume rendering

Instead of rendering the volume by defining a surface first, we can render the volume in a translucent manner, taking into account the brightness of the material beneath the surface. Figure 2.3 shows a volume rendering of a human head, created by using the Julius\(^2\) framework.

One such technique, based on *ray casting* is described in Levoy (1988): surface shading calculations are performed at every voxel with local gradient vectors serving as surface normals. Surface classification operators are then applied to obtain a partial opacity for every voxel. The resulting colour and opacity of each pixel in the image plane is blended from back to front by tracing rays from each pixel on the image plane. The result is a semitransparent, three-dimensional image of the volume. This technique is particularly suitable when amorphous phenomena such as clouds, fog and fire are represented.

\(^2\)A software framework for the development of medical visualisation tools. More information can be obtained from their website at [http://www.julius.caesar.de](http://www.julius.caesar.de)
2.3 Reducing storage demand

Images and volumes need to be stored in computer memory to be processed. The computer’s memory is a limited resource.

We define a *byte* to be a unit of storage containing 8 *bits*, each representing either the value 1 or 0. We will use the tradition within computer science to describe a *kilobyte (KB)* as exactly 1024 bytes, a *megabyte (MB)* as $1024^2$ bytes and a *gigabyte (GB)* as $1024^3$ bytes.

The number of voxels in an isotropic volume increases rapidly with increasing resolution. While a large, uncompressed two-dimensional image of dimensions $4096 \times 4096$ and 16 bits to represent each pixel's intensity value needs $4096 \cdot 4096 \cdot 16$ bits = 32MB of storage, a three-dimensional image with the same resolution along the z-axis requires $32$MB $\cdot 4096 = 128$GB of storage.

The latter size is significantly larger than the size of the working memory found on commodity computers today, which typically allows a maximum of 4GB addressable primary storage memory. In modern memory architectures, primary memory is typically divided into several stages where smaller and faster memory sizes operate at a higher speed than larger, slower memory types. This process is called *caching*. Additionally, most operating systems allow large data to be seamlessly swapped in and out of working memory in a process called *swapping*. Both these processes are based on spatial locality in the data to be accessed, in order to keep the number of accesses to slower larger media to a minimum. It is therefore necessary for a storage-demanding algorithm to exploit spatial locality in order to achieve maximum efficiency.

In our case, we seek to process 8-bit isotropic volumes of size $2048 \times 2048 \times 256$, which require $3^{1.0}$GB of storage. When processing an image it is common that more data than just the image need to be stored in the computer’s memory at some point in the process, making the effective requirement somewhere above 1.0GB. Based on this we have investigated means of reducing storage requirements in the following sections.

2.3.1 Compression of volumetric data

Because the size of the raw data makes them difficult to handle, it would be desirable to compress them so that they require less storage space. We will therefore examine some common methods for the compression of voxel structures in this section.

---

3Although volumes can have a larger number of xy-planes than 256 (slices), the original volumes are split into N volumes of 256 slices each to facilitate transport of the data.
2.3. Reducing storage demand

All kinds of data compression can be divided into two principal groups: those that are information-preserving and those that are “lossy” (Sonka et al. 1999). Information-preserving compression enables error-free reconstruction, while lossy methods will introduce some form of error. The compression obtainable through lossy compression tends to have substantial advantages with little or no visible impact on the data. However, we consider them unsuitable for scientific use, because they may affect results in an unpredictable manner.

Octrees

A popular, information-preserving method for storing voxel data is the octree. Octrees achieve image compression by storing information in a hierarchical tree structure. This structure is built by recursively subdividing inhomogeneous regions of the volume into eight sub-regions. This procedure is repeated until each terminal node of the tree corresponds to a region of the volume, in which all voxels are uniform or connected. This technique is most efficient when the data is sparse, and large homogeneous regions are present (Samet and Webber 1988).

One disadvantage of an octree is the time it takes to construct the tree. Additionally, the tree must be reconstructed whenever we want to change or add something to the structure. Even small variations on the input can result in a very different tree (Samet and Webber 1988). The time used to maintain the tree structure, can be reduced significantly by keeping changes to the structure in a queue, and only periodically rebuild the actual tree structure. Any algorithm requiring mostly read access to its data will probably benefit from using this structure.

2.3.2 Implementing space-efficient image information storage

This section formalises our mentioned storage demand problem (see Section 2.3) and suggests how to solve it.

There are primarily two types of data that we want to access.

1. 8-bit volumes (originals)
2. Binarised volumes (processed)

Segmentation is the process of separating different parts of an image into different classes. In the case where the segmentation results in two classes only, the result is called a binarisation of the image.
Binary data is the result of applying the binarisation process described in Chapter 3 to original tomographic image data. The original data sets are roughly $10^9$ voxels in size, c.f. Section 2.3. The size may vary somewhat so we will use it as an estimate only.

This distinction is useful because the original volumes may be processed by a high-end workstation, but the binary images should be usable to scientists using commodity workstations.

Storing 1.0–2.0 GB of data is within reasonable limits on modern graphical workstations. This makes cache optimisation the most critical motivation for selection of an efficient storage structure. The simplest way to store such data is in a three-dimensional array, indexable by the grid coordinates $x, y, z$. This will effectively result in a linear representation on disk, where consecutive image cross-sections (slices) are stored as a series of horizontal rows.

If the particular algorithm requires access to a local neighbourhood around a single voxel element, many cache misses will occur because data from different image slices are spread out on disk. Interleaving image slices or splitting up the data in smaller cubic structures may increase cache efficiency.

If the whole volume does not fit in the workstation’s memory, the data must be separated into smaller units, such as image slices or smaller blocks called pages. We then load as many pages as possible, and load and unload pages on demand. This is commonly called .

**Storing the binary data**

Binary data can be stored using only 1 bit per pixel. Because the smallest addressable unit on most modern CPUs is a single byte, we must pay for this reduction in storage space by adding extra complexity to the access of each individual voxel. However, because more data will fit into the faster cache memory, it can potentially make calculations faster.

Extra calculations are required to isolate a single bit whenever a voxel is accessed. However, this enables us to store 8 times as much data in the same amount of memory.

The particular mapping from eight voxels to a single byte must be decided upon. The mapping should exploit spatial locality to utilise the computer’s cache for improved performance.

We have proposed two different mappings for this purpose in more detail.

1. A cube consisting of $2 \times 2 \times 2$ voxels mapped onto a single byte
2. Let bits increase along one of the coordinate axis, for example for each 
(x,y). The value of (x,y,z) is stored for 8 successive z-values.

The first has the advantage that it can be directly used to represent levels in 
an octree. The second has the advantage of having several slices in parallel 
accessed at the same time. Because data are stored linearly in memory, 
this mapping may increase cache efficiency whenever data is accessed across 
several volume slices. This is the case when spatial averaging over adjacent 
voxels is applied.

If the dimension of the total data is not an exact multiple of 8 in either 
direction, it is necessary to pad out the extra data with zeros.

The two mappings mentioned are described in more detail in the following 
sections.

2-cube mapping. The 2-cube mapping is done by first finding the cube 
which stores the voxel value and then finding the value (bit) of interest in 
the particular cube.

This is the mapping from pixel coordinates to 2-cube coordinates:

\[
x_{\text{cube}} = \left\lfloor \frac{x}{2} \right\rfloor \tag{2.1}
\]

\[
y_{\text{cube}} = \left\lfloor \frac{y}{2} \right\rfloor \tag{2.2}
\]

\[
z_{\text{cube}} = \left\lfloor \frac{z}{2} \right\rfloor \tag{2.3}
\]

When the 2-cube that holds the pixel is found by equations (2.1) through 
(2.3), a mapping from pixel coordinates to bit number must be done. This 
mapping is given by equation 2.4, where mod is the remainder operator, 
modulo.

\[
pos(x,y,z) = 4(x \mod 2) + 2(y \mod 2) + (z \mod 2) \tag{2.4}
\]

The mapping is visualised in figure 2.4.

8-vector mapping. As for the 2-cube mapping, the vector (byte) con-
taining the pixel must be found first, and then the bit-number of the pixel 
must be found.

This is the mapping from pixel coordinates to 8-vector coordinates:

\[
x_{\text{vector}} = x \tag{2.5}
\]

\[
y_{\text{vector}} = y \tag{2.6}
\]

\[
z_{\text{vector}} = \left\lfloor \frac{z}{8} \right\rfloor \tag{2.7}
\]
When the 8-vector that holds the pixel is found by equations (2.5) through (2.7), a mapping from pixel coordinates to bit number must be done. This mapping is given by equation 2.8.

\[ \text{pos}(x, y, z) = z \mod 8 \]  

(2.8)

## 2.4 Improving performance

Due to the great amounts of data associated with tomographic imaging, efficient processing is critical. Processing time can be reduced by solving parts of the image processing problem in parallel. Parallelism is applicable to many problems and new technologies have made parallel computation available even on desktop workstations. In light of this we focus on parallel methods of computation in this section, in addition to native optimisations which is a field that has seen some changes after the introduction of cross-platform code that newer high level languages, like Java produces.

### 2.4.1 Parallel and distributed computing

Any algorithm working on pixels or regions independently of each other, can be solved in parallel. Both the Marching Cubes algorithm and SUSAN filtering (see Section 3.2.2) can be computed in parallel. The parallel computation can be done by a computer with several processing units or several inter-connected computer.

It may be possible to partition the data in a way that better enables parallel computing. A simple solution is to process each slice (or a number of slices) of the volume as a separate image. Another possibility is to order the data in a way that guides the traversal.
Another way of exploiting parallelism is to use the SIMD\textsuperscript{4} capabilities available on many modern processor architectures, including the popular Intel Pentium IV and compatible processors. Traditional computers put just a single data element into a register, even if there is room for several data elements. A SIMD system packs multiple data elements into a single register and performs the same calculation on all of them at the same time.

In addition to SIMD processor extensions, the development of cheap programmable GPUs\textsuperscript{5}, have sparked the scientific community’s interest. Modern GPU units do not have a fixed pipeline, and nearly every aspect can be reprogrammed. Programmable GPUs can process data in a way that is equivalent to the SIMD principle described above (Bajajy et al. 2004).

In Moreland and Angel (2003), GPU capabilities are used to make an efficient implementation of the Fast Fourier Transform algorithm. Bajajy et al. (2004) demonstrates that it is possible to accelerate the calculation of solutions to any linear system by using SIMD technology.

2.4.2 Native optimisations

\textit{Java} is a high-level interpreted programming language. This means that instead of compiling executable code for a particular platform, code is interpreted into native code during execution by the Java interpreter. Java is widely used due to its simplicity and cross-platform compatibility. Even though the performance of Java-based applications has steadily improved since its introduction, there are still possible performance benefits to be gained from using natively compiled code. Another important restriction is that it is impossible to explicitly access platform specific benefits, such as SIMD instructions.

The Java Native Interface\textsuperscript{6} (JNI) allows integration of natively compiled code with Java. We have used JNI for our programs, see Appendix A for more information. This enabled us to interface and integrate our software with the existing Java-based framework\textsuperscript{7} as well as give us access to platform specific benefits.

\textsuperscript{4}Single instruction Multiple Data
\textsuperscript{5}Graphics Processing Unit
\textsuperscript{6}See \url{http://java.sun.com/j2se/1.5.0/docs/guide/jni/}
\textsuperscript{7}PFI standardizes on Java
2.5 Conclusions

We suggest that volume rendering should be considered as an alternative to the existing methods based on surface rendering, as it may convey interesting aspects of the data.

When working on grey-level volume data, attention must be paid to how cache and page faults impact performance. Interleaving several image slices or reorganising the data into smaller cubic volumes may improve spatial locality of the data. Testing should be performed in order to evaluate its impact on performance.

When working on static binary data, an octree should be used for more compact storage. This is relevant for both visualising and analysing the data. When data is binarised, a compact storage structure representing a voxel by using only a single bit should be used.
Chapter 3

Binarisation

This chapter will present and evaluate the most current methods (Holmstad 2004, Holen and Hagen 2004) for fibre image binarisation used on absorption mode tomographic images, as well as suggest improvements to them and test the effect of these.

3.1 Introduction

Most of the feature extraction methods assume that image data are binarised into material and void, c.f. Chapter 4. This is sufficient for the examination of structural and transport properties, which is mainly concerned with the porous phase\(^1\), c.f. Chapter 4.

Additionally, the binarisation into material and void is useful as a starting point for segmenting a volume image of paper-fibres into individual fibres (Holen and Hagen 2004).

Due to noise, fuzziness and artifacts introduced as part of the image acquisition process, the conversion from a noisy grey-scale image to an accurate binary representation of the fibres is not trivial.

3.2 Current method of binarisation

This section will explain the method used in Holmstad (2004, p. 100) for binarisation of volumes of fibrous structures. A good understanding of the

\(^1\)A phase is a physically distinct portion of matter in a non-homogeneous system. The porous phase is the phase of the pores in a porous material.
current method is believed necessary in order to improve on it or alter it to work on volumes of other materials.

The binarisation process can be divided into two parts – the part done at the facility for image acquisition, ESRF, referred to as off-site image processing, and the part we control directly, referred to as on-site image processing.

### 3.2.1 Off-site image processing

Normalisation and down-sampling are performed off-site.

**Normalisation and down-sampling.** According to Holmstad (2004, p. 96) the intensity values in the volume resulting from the image acquisition method (see Chapter 1) are real values stored as 32-bit floats. To reduce storage demand and facilitate transport, a conversion to 8-bit integer format is performed. A histogram equalisation is first performed in order to maximally stretch the contrast. The results are then down-sampled to 8 bits. The equalisation ensures that the full dynamic range available in 8 bits is utilised.

### 3.2.2 On-site image processing

Images are first rotated and cropped. A smoothing of the images is done to reduce the effect of noise. The images are then binarised into solid and void by thresholding and isolated single pixels are removed. Each step is detailed below and associated image processing techniques are explained.

**Rotation.** The volume is rotated so that material height i.e. its thickness is aligned along one of the axes. The rotation angle is found using Sobel operators. The rotation itself was performed by voxel-transformation.

**Cropping.** *Cropping* is applied manually after the rotation.

Cropping is a simple procedure where a subsection, or a bounding box or cube of the gathered data volume, is selected so that the region of interest, e.g. the fibre structure, is entirely contained within the volume. Cropping should be applied as early as possible so that the amount of data to be processed is lessened in later procedures. Automatic cropping can easily be applied on the final binarised volume, by detecting the volume’s bounding cube.
3.2. Current method of binarisation

**Edge preserving noise removal.** The volume is smoothed by averaging all slices with the neighbouring slices right before and right after in sequence. Holmstad reported that this smoothing removed much of the small-scale noise, but made the grey-scale transition over phase borders slightly more washed out.

Holen and Hagen (2003) suggested an improved method of smoothing that preserved phase border, i.e. edges were not blurred. Two promising methods for edge-preserving noise reduction was investigated as a result of a survey of the literature in the field, namely the “Smallest Unvalue Segment Assimilating Nucleus” (SUSAN) routine and anisotropic diffusion. Holen and Hagen (2003) concluded that the results from using the two methods are close to equal. The Susan routine was applied per slice, disregarding information in the neighbouring slices.

The SUSAN routine was invented by Smith and Brady (1997), who found it to provide a good quality to speed ratio compared to competing routines. The SUSAN principle can be used as a basis for feature-detection and structure-preserving noise reduction. The structure is preserved by only smoothing regions with low variance in intensity. It is in essence a form of adaptive neighbourhood smoothing, as described in Sonka et al. (1999).

**Global thresholding.** The volume is globally thresholded with a manually selected threshold value.

*Thresholding* is a method of segmentation. Basic thresholding is done by visiting each pixel (or voxel) in an image, and set the pixel to $v_1$ if its value is above or equal to a given threshold value and to $v_2$ if the threshold value is below the pixels value, where $v_1$ and $v_2$ are constants denoting certain intensity values. The values $v_1 = 255$ and $v_2 = 0$ are often chosen.

There are two main types of thresholding,

- **Global** – the image is thresholded with a constant threshold value
- **Adaptive** – the threshold value of a pixel depends on the intensity values in the area around that pixel (optionally including that pixel’s value)

Both global and adaptive thresholding-algorithms can in principle be fully automated.

In the discussed procedure a constant threshold value is manually selected and applied globally.

**Glue and capillary removal.** The capillary which the material sample is glued to should be removed from the volume in addition to the glue itself,
as it is no part of the fibre structure. The glue and capillary can be seen in figure 1.2.

The glue and capillary was removed after the thresholding by a watershed (Sonka et al. 1999) routine. Holmstad reports that the watershedding was done in several steps to remove both the capillary and the glue while attempting to preserve the fibre phase. The approach succeeded in removing the capillary completely, but the glue segmentation was only moderately successful. Using other types of glue or other methods of mounting may allow for more successful glue-removal in volume images. The capillary and glue should be removed before applying auto-threshold. The existence of the capillary and glue will affect the histogram, by effectively adding two distributions (capillary and glue) to it.

Small region removal. Regions smaller than $216 \mu m^3$ are removed.

Holmstad found removing regions smaller than $216 \mu m^3$ to be a good trade-off between preservation of structural features and removal of noisy elements from binarised volumes of paper.

The voxel-count of a region can be found using the flood-fill algorithm, where all connected voxels in a region are marked as belonging to that region. Regions with a voxel-count less than $216 \mu m^3/v$, where $v$ is the volume of a single voxel, can then be removed.

3.3 Discussion of current method

We have examined the current method for binarisation of volumes resulting from absorption mode tomography critically. This section gives our findings in this respect.

The equalisation and bit-reduction performed off-site may introduce inconsistencies between samples taken from different materials.

Consider tomographic images of two materials with different X-ray absorption properties, resulting in different contrast in the images, while the noise is of the same character. After equalisation they may end up having the same grey-level values, even though the material’s intensity values originally was in different ranges.

This procedure may complicate direct comparison of different materials.

The voxel-rotation can seriously degrade image quality. Cubic voxels can only be rotated in $90^\circ$ intervals without introducing some form of data distortion, c.f. Section 2.1.1. All other rotations will inevitably require some
form of interpolation to map the rotated volume onto a new one. The effective resolution will be reduced because of this. The degree of loss depends on the particular angle, and it is difficult to assess exactly how much distortion is introduced. As far as we can judge, the impact of this distortion has not been properly acknowledged in current research. One of the underlying assumptions for all the feature extraction parameters introduced in this report, is that voxels are isotropic. We will therefore strongly suggest that alternative methods for either sample mounting, or reconstruction is further investigated to at least lessen the impact of voxel rotation on the acquired data.

The currently best pre-processing is done by analysing two-dimensional image slices independently throughout the volume (Susan-filtering of each slice). Pre-processing that uses three-dimensional information has the potential to result in higher quality results, because for each action chosen, more information is potentially used to guide that particular choice. Based on this, we recommend that the pre-processing is extended to use information from neighbouring slices. An extension of Susan filtering to three dimensions is described by Skocir et al. (2002).

Threshold value selection can be done automatically, possibly with even better results than that resulting from a manual selection. This property is easily confirmed by noting that for a high-resolution volume, e.g. \(1000^3\) voxels, a human observer will have a hard time verifying which threshold value is the best one, although she might quickly find the range where the optimal threshold value lies and narrow the choice down to a few values.

Removing small regions as explained in the previous section could possibly be a bad choice when using the binarised volume as a means to reach an individual-fibre segmentation, as in Holen and Hagen (2004). Additionally, the connectivity-criteria used when removing regions or pixels can be of significance to the results. Tests should be performed to examine these possibilities. Due to time constraints we have not performed these tests.

\section*{3.4 Suggested improvements}

Based on the discussion in the previous section, we suggest some improvements to the current method of binarisation of absorption-mode computed tomography volumes. As in the treatment of the current method of binarisation, this section is divided in two — off-site and on-site processing.

Our improvements does not change the procedure steps, it changes the actions done in those steps. The changes should be viewed as incremental improvements, as they do not change the process suggested by Holmstad
(2004) to a large degree. Only the steps where changes are suggested are brought up in this section.

3.4.1 Off-site processing

Higher quality data can be expected if voxel-rotation is avoided, c.f. Section 3.3. Based on this, it would be best if a correction angle was used in the Radon transform, rotating the raw data, in order to limit the distortion the rotation introduces to a minimum.

Another option, less favourable, but still an improvement, would be to voxel-rotate the Radon-transformed data before the data is down-sampled, and use high-order interpolation of pixel values in the rotation process.

Yet another option is to solve the problem of rotation practically, by making changes to the equipment at ESRF, so that volumes do not get random orientation as a result of the image acquisition process.

3.4.2 On-site processing

This section gives a detailed description of the proposed new on-site method of processing.

Rotation. The need for voxel-transformations should be eliminated by performing the rotation at ESRF on the raw data, as described earlier in this section.

Improved edge preserving noise removal. The Susan noise-removal can easily be extended to use information from a third dimension, as shown by Skocir et al. (2002). The only changes are that the mask moved around in the image is three-dimensional and the arithmetic on the pixels covered by the mask takes this change of mask-properties into consideration.

As reported in Smith and Brady (1997), Susan filtering can be iterated for enhanced smoothing, and in the tests presented, multiple iterations of Susan filtering did not degrade edges or corners, but did significantly reduce variance in the image. In Holen and Hagen (2004), a $\sigma$ of 4 followed by a second iteration with a $\sigma$ of 2 was applied. The running time of the three-dimensional Susan algorithm is roughly proportional to $\sigma^3$, hence if we can achieve the same or a better result using several iterations with a smaller $\sigma$, we will be able to reduce the time usage of the algorithm.

The implementation of Susan noise-removal used is detailed in Appendix A
Improved global thresholding. When there are two grey-level distributions present in the image, a statistical optimal threshold value that separates the distributions can be found automatically. This binarisation is commonly thought of as dividing the image into foreground (objects) and background.

A classic and quick way of finding the optimal threshold value is to use the method “Calvard and Ridler optimal thresholding”. The algorithm is given below (Sonka et al. 1999):

1. Assuming no knowledge about the exact location of objects in the image \( f \), consider as a first approximation that the four corners of the image contain background pixels only and the remainder contains object pixels.

2. At step \( t \), compute \( \mu_B^t \) and \( \mu_O^t \) as the mean background and object grey-level, respectively, where segmentation into background and foreground at step \( t \) is defined by the threshold value \( T^t \) determined in the previous step.

\[
\mu_B^t = \frac{\sum_{(i,j) \in \text{background}} f(i,j)}{\# \text{background pixels}}
\]

\[
\mu_O^t = \frac{\sum_{(i,j) \in \text{foreground}} f(i,j)}{\# \text{foreground pixels}}
\]

3. Set

\[
T^{(t+1)} = \frac{\mu_B^t + \mu_O^t}{2}
\]

\( T^{(t+1)} \) now provides an updated background-object distinction.

4. If \( T^{(t+1)} = T^t \), halt, otherwise return to step 2.

The first step in the algorithm is sometimes changed to use a random starting value for the threshold. As an improvement to the algorithm above, the image histogram can be searched instead of the entire image, the \( T^t \)s can be stored as real numbers and the comparison in the last point be changed to ‘If \(|T^{(t+1)} - T^t| < \epsilon\), halt, otherwise return to step 2.’, where \( \epsilon \) is the tolerance, a low value like e.g. 0.01. If this latter improvement is used, care must be taken to avoid an infinite loop when the difference between the \( T \)s is not converging, but oscillating in a range larger in size than \( \epsilon \).

The most recent improvement on this method and its likes was done by Lin (2003). Lin presents a new method that finds the optimal threshold by solving a nonlinear equation describing the zero derivative of the between-class variance of the image. Lin’s algorithm supports both histograms with a
single peak and with multiple peaks, in addition to avoiding the rounding errors and possible premature convergence associated with the Calvard-Ridler algorithm (Lin 2003).

The methods above will inevitably fail to correctly classify all parts of the image in cases where the distributions in the image overlap. If so, image thresholding by Kriging is an option (Oh and Lindquist 1999). This method takes into consideration that the distributions overlap and decides on a range of overlap in intensities and use a special method called Kriging to correctly classify the pixels with intensities in this range.

3.5 Preliminary results and discussion

This section shows the effect of using our suggested improvements. A test on using the three-dimensional Susan instead of doing the two-dimensional Susan per slice is done, as well as a test showing the same data processed by a simplified version of Susan algorithm for ten iterations. Both tests will be based on the Calvard-Ridler optimal thresholding.

When comparing two different methods of noise-reduction filtering, synthetic images with artificially added noise should be used as the basis for comparison. This will provide a known reference image with known noise. A simple comparison can then be done by calculating how close the filtered image is to the original image, or to see to which degree the important features in the original image were reconstructed from the noisy image. More advanced signal-to-noise ratio calculations can also be performed. Unfortunately, the noise in our images is of an unknown nature, and we do not have a noise-free reference image.

If we seek smoothness, then the amount of variance in the image’s intensity values can be used as a metric on quality. However, it is not given that the technique with the lowest associated variance in result-images is the better one, e.g. edges could be more blurred. Wilson et al. (1999) compared several noise-reduction filtering techniques, and found that the ones with the lowest variance were not the best, and concludes that it is imperative to include human-observer models and experiments in the analysis of noise-reduction filtering of noisy image sequences. Based on this, we will make use of visual inspection to measure the quality of noise-reduction. We will also make use of the information the variance gives in these measurements.
3.5. Preliminary results and discussion

3.5.1 Test of improved noise-removal

We have suggested two possible enhancements to the existing method based on Susan filtering. The first is to perform filtering by using a spherical three-dimensional mask rather than a circular mask. The second suggestion is to use a multiple application of a smaller cubic mask, referred to as Fast-Susan, to increase the speed of the processing.

The comparison is done by running all filters on the same image volume. The viewing axes are rotated $90^\circ$ after filtering, so that we can see the changes resulting from taking three-dimensional information (depth) into account visually.

The mentioned FSL-based implementations of Susan2D and Susan3D was used for this test. The flipping of axes was done in ImageJ by selecting “image->stacks->reslice->Rotate 90 Degrees + Start at Left” from the menu. In addition to this, we selected slice number 65 for presentation, near the middle of the volume.

Ten consecutive iterations were applied using the Fast-Susan implementation. The choice of ten repetitions was somewhat arbitrary, because several more iterations could have been performed until the algorithm reached stability. Unfortunately a certain amount of leaking between fuzzy borders occurs, and the process must therefore be stopped after a limited amount of iterations. The value ten was considered a good trade-off between smoothness and leakage.

A volume-image of paper-fibres acquired through X-ray synchrotron tomography was used for the test. This volume will be referred to as $S1^2$, and is of the size $100 \times 100 \times 100$. The results are given as three image series, which consists of five images each. The series are (in order presented):

1. Original volume (figure 3.1)
2. Volume rotated $90^\circ$ (figure 3.2)
3. Global threshold based on Calvard-Ridler (figure 3.3)

The images in each series are (in order presented):

a. Original image slice
b. Image slice filtered twice using Susan2D with $\sigma = 4.0$, first with brightness threshold (bt) 40, then with bt 20 as in Holen and Hagen (2003). This is included for comparison

c. Image slice filtered with same parameters as above, but using Susan3D

$S1$ is a sub-volume of the volume known as $S8XZ$ internally at PFI.
d. Image slice filtered with Fast-Susan2D 3x3 mask ten times with bt 40

e. Image slice filtered with Fast-Susan3D 3x3x3 mask ten times with bt 40

Figure 3.1: Sample from original volume

The images shown in figure 3.1, are shown in the original direction of the smoothing. There is practically no difference between the two- and three-dimensional filtered images. Images d, and e shows more clearly defined regions, but all significant edges are kept from the original image.

In figure 3.2, we have rotated the volumes shown in figure 3.1, so that we can more clearly see the effect of using the information from consecutive slices. There is significantly more noise in the two-dimensional filtered images. This noise is due to the independent smoothing of consecutive image slices. The difference is most apparent when images processed by Fast-Susan are compared. We see that the three-dimensional version shown in e has a better separation between the regions. In areas where the edge is not properly defined we notice a certain amount of “leakage” between the regions.

Table 3.1 shows the mean and variance of the original and filtered versions of the S1 volume, where the suffix X10 means the filter was applied ten
3.5. Preliminary results and discussion

Figure 3.2: Sample from volume rotated $90^\circ$

times consequently. Note that the variance is slightly lower in Susan3D than Susan2D, but the variance in Fast-Susan3DX10 is higher than in Fast-Susan2DX10. We believe the first observation is best explained by Susan3D doing a better job than Susan2D at removing noise, i.e. produces more smooth volumes. However, the latter observation conflicts with this theory if interpreted the same way. Another explanation is that Fast-Susan2DX10 indeed produces more smooth volumes than Fast-Susan3DX10, but the increase in smoothness comes from smoothing across edges, breaking region boundaries. This theory is strengthened by visual inspection. Figure 3.2(d) shows an example of this. The lower, central part of the contour of the white fibre in the lower half of the image has a more blurred region border than the one in figure 3.2(e).

A single threshold value was calculated for each volume, based on the Calvard-Ridler method. This was applied to optimally binarise each volume. The implementation of Calvard-Ridler in ImageJ was used for the test. This implementation uses a random start threshold, compared to the algorithm in Section 3.4.2. The values it found was: a:97, b:94, c:94, d:89 and e:90 for the rotated volumes.
Table 3.1: Mean and variance of original and filtered volumes

<table>
<thead>
<tr>
<th>Image</th>
<th>Mean</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>68.763</td>
<td>42.763</td>
</tr>
<tr>
<td>Susan2D</td>
<td>67.120</td>
<td>34.946</td>
</tr>
<tr>
<td>Susan3D</td>
<td>67.097</td>
<td>34.584</td>
</tr>
<tr>
<td>Fast-Susan2DX10</td>
<td>62.962</td>
<td>33.888</td>
</tr>
<tr>
<td>Fast-Susan3DX10</td>
<td>62.886</td>
<td>34.149</td>
</tr>
</tbody>
</table>

Samples from the binarised volumes are shown in figure 3.3. The samples correspond to the ones in figure 3.2. There is little difference between the images b and c, so we cannot conclude from this test that the binarised results were significantly affected by only applying three-dimensional smoothing in the previous step. However the difference between images d and e indicates that there are some benefits from using a three-dimensional mask when more iterations are applied. The difference is most visible when comparing continuous horizontal edges in the image.

Arguably the result in image e is considerably less noisy and better defined than the others. We also see that the perturbations resulting from leaking in fuzzy edges as seen in figure 3.2 have been surprisingly well defined after thresholding.
3.5. Preliminary results and discussion

Figure 3.3: Sample from auto-thresholded volume


3.5.2 Time usage

<table>
<thead>
<tr>
<th>Variant</th>
<th>Estimated pixels per second /1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susan2D</td>
<td>8.8</td>
</tr>
<tr>
<td>Susan3D</td>
<td>0.47</td>
</tr>
<tr>
<td>Fast-Susan2D</td>
<td>87</td>
</tr>
<tr>
<td>Fast-Susan3D</td>
<td>54</td>
</tr>
</tbody>
</table>

Table 3.2: Time usage of different Susan-variants, \( \sigma = 4.0 \)

The different variants of Susan (2D/3D - fast/regular) have large differences in processing time. The time usage has been measured by running the variants on the same, large image volume (slices 10–50 from S1), and retrieving the number of pixels processed per second from ImageJ reports (table 3.2).

3.5.3 Summary and discussion

Based on the test above we conclude that simply extending the two-dimensional Susan filter to three dimensions and repeating the process using the same parameters, did not significantly improve the binarised result. However, we successfully demonstrated that several iterations using the Fast Susan filter in three dimensions generated more clearly separated regions in the final binarised volume. This does not imply that similar results could not have been achieved by running the same amount of iterations with a large \( \sigma \). However, considering the difference in time usage, we consider it to be a significant improvement to the current process.

The automatic threshold found using the Calvard-Ridler method resulted in good separation between regions. We will limit ourselves to claim that the automatically-found threshold was not visibly worse than those we found by visual inspection, and as such can be used in the binarisation process.
Chapter 4

Feature extraction

This chapter aims to give an overview of existing measures done on volumes of paper samples and give the reader good references to details on the covered subjects. Two new ways of measurement not covered in the paper science literature are introduced, namely fractal analysis and the material orientation. A discussion of practical application of such measurements is included at the end of the chapter.

4.1 Existing methods

This section presents the existing methods available for measurements on fibrous structures. The presentation is based on (Holmstad 2004), which is the most recent study on paper structure characterisation known to us. We have given a summary of these methods.

4.1.1 Surface area

The surface area is a general property that can be measured on all materials. It can be approximated by counting the number of voxel edges between the porous phase and the solid phase and multiplying with the physical area each voxel represents. In Huang et al. (2002), this counting was done by for each voxel in the fibre phase check if that voxel has one (or more) void phase voxels in its 6-neighbourhood, if so, increase the count.

Due to the cubic size of the voxels, this approximation will only give an upper bound on the actual surface area. Better estimates can be made by the more complicated procedure of polygonizing the data in such a way that corners are smoothened, and then summing the polygon area (Holmstad 2004).
The specific surface area is a common used measure, and is the ratio of the surface area to the volume of the sample.

According to Holmstad (2004, p. 20), specific surface area measured on image slices has shown a good correlation with the scattered light coefficient.

### 4.1.2 Void fraction (porosity)

Porosity is a commonly used measure for describing porous materials. Consider a sample of total volume $V$. Define the volume of the solid phase to be $V_s$, and the volume of the pore phase (the holes) to be $V_p$, with $V = V_s + V_p$. The volume fraction is a normalised variable that is generally more useful. The volume fraction of the pore phase is commonly called the porosity, and is denoted $\Phi = \frac{V_p}{V}$. The solid volume fraction is then $1 - \Phi$ (Garboczi et al. 1999).

Holmstad (2004) divides the paper structure into layers of locally uniform height along the height direction, and then calculates porosity for each layer. The surface is defined using a rolling sphere algorithm, as described by Hilpert and Miller (2001). The porosity $\Phi$ for each layer is estimated by the equation below, where $n$ is the phase, $i$ is the layer and $M_i$ is the number of pixels in layer $i$.

$$\Phi_{n}(i) = \frac{|p | p \in n|}{M_i} \tag{4.1}$$

**Thickness**

Thickness can be calculated as the distance between the top and bottom of the paper surface along each vertical line, and each slice of the volume. To define the surface of the paper, a rolling sphere algorithm (Hilpert and Miller 2001) can be applied (Holmstad 2004, p. 24). The calculated thickness will then be dependent on the radius of the rolling sphere.

**Layer density**

Layer density is given by equation 4.2, layers and $\Phi$ are defined as in Section 4.1.2, while $\rho$ is the density.

$$\rho_{layer} = \Phi_{fibre} \cdot \rho_{fibre} \tag{4.2}$$
4.1. Existing methods

Basis weight

According to Holmstad (2004), the weight or substance per unit area is fundamental in paper and paper board products. The basis weight of paper is the weight per unit area, and is given by equation 4.3, where $\beta\ [g/m^2]$ is the basis weight, $T$ and $[m]$ is the thickness, $\rho\ [kg/m^2]$

$$\beta = T \cdot \rho_{solid} \cdot 1000$$  \hspace{1cm} (4.3)

Mean pore size

The mean pore size is the mean length of all vertical or horizontal lines passing through the body of the paper sample, covering only void. Mean pore size can be calculated on both two- and three-dimensional images (Holmstad 2004, p. 108).

Pore chord distribution

Pore chord measurements are a way of assessing information about the geometry of the void phase by measuring the pore chords in the structure. A pore chord is the local distance between two solid elements in the material (figure).

Pore chords can be measured in any spatial direction, and gives a measurement on the local one-dimensional extension of the pore volume. Mean pore height and width can be estimated using pore chords.

When average pore chord length (also called mean free path) is determined for all three principle axis, a visualisation of this information as an ellipsoid, called the equivalent pore ellipsoid (see figure 4.1.2), can be obtained. The equivalent pore ellipsoid has proven useful to describe differences between porous materials (Holmstad et al. 2003, Holmstad 2004). The oblongness of this ellipsoid, or its ellipticity is also known as the structure anisotropy, and it is measured separately in each principal direction.
4.1.3 Pore hydraulic radius distribution

Pore radius is a measure of the average radius of cross-sectional pore areas. Hydraulic radius is defined as the ratio between pore area and the perimeter of the continuous pore area.

The hydraulic pore radius is normally given as an area weighted distribution along a principal direction across the volume. Ramaswamy et al. (2001) defines pore hydraulic radius \( (R_{p,h}) \) by equation 4.4, where \( A_p \) is the cross-sectional area of a pore [\( m^2 \)] and \( W_p \) is its perimeter [\( m \)].

\[
R_{p,h} = \frac{A_p}{W_p} \tag{4.4}
\]

Pore size distributions

Huang et al. (2002) defines a pore as a region that comprises of only interconnected pore pixels within the region, and none of the pixels in a given region is connected to other pore pixels in other regions. The connectivity criteria used by Huang et al. (2002) is 8-connectedness in two dimensions and 6-connectedness in three dimensions.

The area of a pore region can be calculated by counting voxels in that region and multiplying the count with the length of the voxel in each direction in a two-dimensional image. The perimeter can be calculated as the number of pixels that have both neighbours in the region and outside the region times the length of each pixel. This goes for two-dimensional images too. In three-dimensions, the volume of the pore region can be calculated similarly to the area in two dimensions, summing up areas along the height-range of the pore.
Holmstad (2004, p. 111) applies a technique based on mathematical morphology in order to determine the distribution of pore volume. The method is based on the maximal sphere method, where all voxels in the porous region are assigned to the largest sphere inside the region that includes the voxel.

4.1.4 Transport properties

An important characteristic of paper structure is its transport properties. Transport properties are of fundamental interest in relation to ink-absorption, filtering and sanitary uses (Holmstad et al. 2003).

Diffusion tortuosity

Tortuosity is a measure of the complexity of paths through the porous structure. The number of paths depend on porosity, anisotropy, degree of layered structure and other parameters according to Holmstad (2004, p. 112).

More specifically, the tortuosity is defined as the ratio of the average length of a path through the porous structure moving only in the porous phase to the straight line (or the shortest length) distance through the porous structure (Ramaswamy et al. 2001).

Diffusion tortuosity can be estimated by a controlled random walk in the volume, simulating the movement of a particle in the structure, allowing only movement in the pore phase (the void parts). For each movement-step in the three-dimensional structure, all 26 possible neighbouring voxels are considered as the next location for the particle, to simulate diffusion behaviour (Holmstad 2004, p. 112). Several random walks are performed and their average ratio is used as the diffusion tortuosity of the volume.

Flow permeability and flow tortuosity

The movement of liquid particles within the structure can be simulated using high-resolution images. The Lattice-Boltzmann method can be used to simulate flow in porous structures (Martys and Hagedorn 2002).

Flow tortuosity is defined as the ratio of the mean length of flow paths to the thickness of the sample in the direction of mean flow (Holmstad 2004, p. 113). The mentioned flows can be calculated using the lattice-Boltzmann method.

Flow permeability is a measure of the ability of a material to transmit fluids through it. The flow permeability depends on the structural characteristics
of the material, such as porosity, pore surface area and tortuosity of the flow paths. Flow permeability can be calculated using the lattice-Boltzmann method (Holmstad 2004, p. 114).

**Conductivity simulation**

Arns et al. (2001) describes a method for simulating electrical conductivity on microtomographic images:

By using the analogy of an electrical network, one can simulate the material’s resistance to fluid flow. Conductivity simulation is based on a solution of the Laplace equation with charge conservation boundary conditions. The three-dimensional voxel micro structure is first converted into a network of resistors by connecting each pair of adjacent pixels by a simulated resistor.

A potential gradient is then applied in each direction, and the system is relaxed using a conjugate gradient technique to evaluate the field.

According to Arns et al. (2001), the simulation results were in excellent agreement with experimental measures over a wide range of porosity.

**Simulation of mercury intrusion porosimetry**

The mercury intrusion porosimetry (MIP) is a method used for assessing the pore size distribution directly on physical paper samples. MIP can be found by applying a modified version of the largest sphere method (see Section 4.1.3). The intrusion will be limited by the pore necks (Holmstad 2004, p. 116).

Holmstad (2004) claims that a comparison between digitally simulated MIP and physically measured MIP can show the effects of the compression of the paper structure.

**Simulation of diffusivity**

Diffusivity is usually associated with thermal diffusion, or thermal conductivity. Water vapour is also known to be, at least partly, a diffusion-type transport, and the diffusivity can hence have an impact on how the material dries. Diffusivity is a relevant parameter for barrier coated (water resistant), and film laminated paper.

The diffusivity can be found from a Monte Carlo simulation approach, by simulating a form of random walk of the molecules from the centre of cubic structures, to the edges of the volume. This technique has been successfully
applied on low resolution three-dimensional paper volumes (Holmstad 2004, p. 117).

The unit of diffusivity is \( m^2/s \), square meters per second.

### 4.1.5 Skeleton based techniques

The *medial axis transform*, often referred to as a *skeleton* (figure), refers to a simplified representation of the original three-dimensional data where important information about the topology and geometry of the original structure is maintained. The medial axis can be defined by the concept of maximal spheres, or spheres. If we sum up all the centres of all maximal spheres entirely contained within the structure, we get a thinned result consisting only of lines and surfaces. A more precise description is given in Sonka et al. (1999, p. 576).

Different methods of constructing the skeleton can yield different skeletons, and the medial axis may in practice be determined by at least three fundamentally different methods. None of these have proven superior to others, so each group working on three-dimensional medial axes develop their own procedure (Holmstad 2004, p. 118). Holmstad emphasises that while the medial axis can be applied for determining many interesting structural parameters and gain more knowledge of the detailed pore extension the method is dependent on the chosen approach.

Even though no standard procedure has been determined, the examination of skeletons is of some value, as long as the particular method for its determination is thoroughly described. Yang (2001) developed a particular method for the determination of the medial axis, and used this to guide analysis of single fibres, see Section 4.1.6.

Holmstad (2004, p. 192) suggests that skeletons could prove to be useful in the assessment of pore-geometry. A skeleton of the porous phase can be used in the detection of *pore throats*, partitioning of the pore space, and in assessment of porous structure parameters.

When a medial axis of the pore phase is known, where each node correspond to a pore body, there will be a pore throat on each branch limiting the flow to adjacent pore bodies. The determination of the pore neck can be done by finding the narrowest location of this branch. While still a difficult task,
Chapter 4. Feature extraction

the detection of this location is simplified when guided by the medial axis (Holmstad 2004, p. 120).

4.1.6 Measures based on individual fibre segmentation

Most of the properties discussed in this chapter have focused on the properties of the entire bulk material, the sum of all non-void parts. Many fibre characteristics, however, require accurate segmentation of individual fibres.

Different approaches have been tried. Methods based on medial axis skeletons have been examined in Yang (2001, p. 28), giving characteristics of fibre bonding and fibre segments. Both Araonsson (2002) and Holen and Hagen (2004) have successfully segmented individual fibres and extracted properties from these.

A perfect segmentation would, for instance, allow the calculation of total free and bonded surface area in the network. A method for calculating the bonded area based on a fully segmented fibre structure was suggested in Holen and Hagen (2004).

Current methods for single-fibre segmentation, have been able to correctly identify 65–75% of the actual fibre structure (Holen and Hagen 2004, p. 49). One of the limitations of the method presented in Holen and Hagen (2004) is that it is based on tracking the lumen, the hollow inside of each fibre, across image slices. The method is thus not reliable in the identification of collapsed or deformed fibres. It was suggested that techniques using other means than lumen tracking should be combined in order to obtain more reliable measurements. Further research should be performed in this area in order to increase the accuracy of such measurements.

According to Holmstad (2004, p. 122), it is still an open question whether research and computational development will make complete fibre segmentation possible.

4.1.7 Surface properties

According to Holmstad (2004, p. 122) it is possible to measure surface properties like roughness based on the surface topography of three-dimensional material structures. The surface can be defined by applying a rolling ball algorithm to the volume (Holmstad 2004, p. 104). However, because data of close to the same resolution (1μm) can be obtained from more available sources, like laser profilometry, the volumes obtainable by X-ray tomographic techniques are better suited for the characterisation of properties.
related to the bulk material. Surface properties extracted from X-ray tomographic images may still be useful in relating surface properties to interior characteristics.

**Surface roughness**

According to Stoen (2003), surface roughness is an important parameter for describing printing paper and wrapping paper. The roughness correlates with optical properties like gloss and gloss variation. Additionally, it has a large effect on printing ink absorption and the resulting print quality.

Stoen (2003) found the arithmetical mean deviation roughness (commonly denoted Ra) to correlate with dot-raggedness, glossiness, mottling and print sharpness. Ra and other roughness parameters are defined in ISO4287 (2000).

Gary Chinga and Dougherty (2003) developed methods to measure chosen roughness parameters as defined in ISO4287 (2000), including root mean square deviation, skewness and kurtosis, and the number of heights and peaks and the height and depths of these. The local facet direction was detected using Sobel operators. Pore-detection was done by simulating a sphere rolling on the porous surface, similar to as described in Section 4.1.3.

A computer program for doing the measures mentioned above is available at [http://home.online.no/~gary.c/IJ/SurfCharJ.htm](http://home.online.no/~gary.c/IJ/SurfCharJ.htm).

Wang et al. (2004) used surface volume, contact fraction and surface pit distribution as measures of surface roughness, as illustrated by figure 4.1.7.

### 4.1.8 Simulation of optical properties

Glossiness and other important optical properties can in theory be calculated from high resolution three-dimensional data. The interaction of light with a surface can be simulated by tracing the path of individual rays of light as they are reflected from the surface. This method is called ray tracing. While most research in ray tracing has focused on creating photo-realistic looking computer images, these methods have only been loosely based on physical properties. A solution for physically based light simulation using forward ray tracing was presented by Brunborg and Smørgås (2003).

Since the original voxel-volumes consists of cubes, a simulation of light interaction on such volumes, will only yield rays reflected in 90°angles and such simulations will differ significantly from the physical reality. Smooth mathematical approximations of the surface structure can possibly be created.
This is a complex task (Holmstad et al. 2003). Some of these techniques are examined in Section 2.2.

One major obstacle in practical light-simulation from voxel-data is the high computational requirements. Another obstacle is that the current effective resolution, after binarisation of 1–2µm obtainable through X-ray tomography, is significantly higher than the wavelength of visible light 400–700nm. For correct measurements, we would probably need a significantly higher resolution. It is not certain, though, that we require perfect representation of the surface in order to measure statistical properties of the light. Holmstad (2004, p. 123) did note perform optical measurements due to the crudeness of the data. It still remains an open question whether practical measurements will become possible in the future, and further research is necessary.

4.1.9 Simulation of network behaviour under physical influence

Future work may enable us to evaluate physical properties based on the fibrous structure in a three-dimensional volume. All current methods assume that the fibre network is unaffected by its surroundings. To quantify such parameters, we would have to obtain data for materials under different physical influence. An example is to measure the same material sample, before and after it has absorbed water. Based on the knowledge obtained from such measurements, models can be developed that enables the simulation of
4.2 New methods

This section presents new methods for measurements on fibrous structures, fractal analysis and the material orientation.

4.2.1 Fractal analysis

In several papers presented at APPITA\textsuperscript{1} conferences in 2001–2003, fractal dimension was used to describe the spatial distribution of grammage, a metric measure of paper weight. The structure and other properties of lignin, a product from wood, is described using fractal dimension in several papers presented at EWLP\textsuperscript{2} 2000 and 2002, the 10th and 11th “International symposium on wood and pulping chemistry” and the “Fifth European workshop on lignocellulosics chemistry for ecologically friendly pulping and bleaching technologies”.

We examine fractal analysis through the use of fractal dimension as a descriptor in Chapter 5.

4.2.2 Material orientation distribution

The material orientations can be found by applying a discrete gradient operator, like approximations to the Laplacian or Sobel masks, to the image volume. Sobel masks can be moved along the main axes to find the gradients in the direction of each axis. This distribution of directions can be visualised as an ellipsoid. The length of the ellipsoid’s main axes corresponds to some measure on the amount and size of vectors resulting from the differentiation along the main axes in the volume.

4.3 A practical application of feature extraction

Holmstad (2004) and Goel et al. (2004) present results on how four important variables applied in the paper-making process affect the resulting material:

1. Calendering
2. Headbox consistency

\textsuperscript{1}Technical Association of Australian & New Zealand Pulp & Paper Industry
\textsuperscript{2}European Workshop On Lignocellulosics and Pulp
3 Addition of reinforcement pulp
4 Addition of a fixation agent

Using analysis of three-dimensional data as the basis for fine tuning these parameters, one can possibly reveal new information on how we can further improve the paper production.

Calendering is a finishing process where the paper is passed through a series of rollers. The compactness of the resulting material will be depending on the amount of pressure applied on the rollers. It was shown that the effect of calendering had a clear effect on most of the measured properties. For instance the porosity, tortuosity, diffusivity and permeability showed considerable differences depending on how much pressure was applied.

The headbox is usually at the front of the paper-making machine and is responsible for the transformation of stock into a machine-wide rectangular uniform flow of pulp at a controlled velocity. The headbox consistency is the ratio of fibre mass to fluid in the output flow (Peel 1999, p. 125). Increasing the headbox consistency had a measurable impact on structural anisotropy and the addition of reinforcement pulp resulted in a significant change in pore chord anisotropy. The addition of a fixation agent were expected to yield a more complex structure, where the increased amount of fine material should somewhat restrict the transport and reduce the size of the pores. These assumptions were also partially confirmed by the results.

The measurements were considered to be moderately successful in the analysis of factorial parameters (Holmstad 2004, p. 176). A problem with the mentioned experiment was that it attempted to assess the effect of several variables on the same data simultaneously. The surprisingly high impact calendering had on the measured properties, possibly due to its significant compression of the void fraction, partly concealed the effect of the other variables measured. However, the results are encouraging and clearly shows that there are practical and economical motivations in the quantification of three-dimensional volumes.

4.4 Relevance to new materials

Because this report was completed with only paper fibre volumes available, it is difficult to assess how well the existing methods are applicable on new composite and absorbent materials. This section will cover our opinion on how well the existing methods are applicable on composite and fibrous materials.

It is immediately noticeable that the existing methods are dependent the
void fraction to a large degree. Additionally, structural properties are relevant for all materials, but the ones depending on the porous phase is not applicable to non-porous materials. In composite materials fibres are introduced into an existing massive material such as plastic in order to enhance different properties of the original material. Ideally, hardly any void fraction should be present, and all features measured will be depending on intermaterial properties. The accuracy of such measurements are dependant on being able to clearly separate the materials in the resulting image data. It remains to be seen if and how this segmentation will be performed.

Super-absorbent materials such as in sanitary binds will have a significant void fraction, and transport properties are probably useful. Since these materials expand significantly when in contact with water, they should be examined both in wet and dry condition. Existing models for diffusivity and transport tend to assume a static fibre network when performing measurements. More complex models may be required in order to simulate how the material behaves under expansion.

4.5 Discussion and future work

We have some research ideas when it comes to measurements on images of material samples and their usage. This section presents those ideas.

**Single-fibre segmentation.** Holen and Hagen (2004) presented a method for single-fibre segmentation, based on lumen tracking, that was able to correctly identify 65–75% of the actual fibre structure. The method was not reliable in the identification of collapsed or deformed fibres. Holen and Hagen (2004) suggested that techniques using other means than lumen tracking should be combined with lumen tracking to obtain more reliable measurements. Yang (2001) used the fibres’ skeletons to track the fibres. The combination of (Yang 2001) and (Holen and Hagen 2004) could possibly lead to a higher fibre-identification rate than with previous methods.

**Using neural networks for function learning.** A neural network is a set of simple computational units that are highly interconnected. The units loosely represents the biological neuron, a cell-type that is associated with learning and thinking in animal (and human) brains. Neural networks can be used to learn, and thereby find, functions based on examples of input to the function and output of the function given those inputs. It has been proven
by Cybenko (1989) that neural networks\(^3\) can approximate any continuous function to any degree of accuracy. The neural network will make a mapping that minimises error in the outputs for the given inputs and the learning is properly administered, the net will generalise from the training vectors to the general case, producing a general mapping from all inputs to all outputs.

If it is believed that multiple microscopic parameters affect a certain macroscopic parameter, the mapping from the microscopic parameters to the macroscopic parameter is desirable to learn in order to predict or control the macroscopic parameter. Such a relation can be complex and thus difficult to obtain by traditional methods like regression analysis.

Examples or “training vectors” that show correct input (microscopic parameter values) and output (macroscopic parameter value) given the input, must be available in order to train the neural network to act correctly.

**Fractal analysis.** The concept fractal analysis needs to be investigated further in order to put it to use. The mathematics behind the fractal dimension and algorithms for estimating it are of special interest in that regard. These topics are treated in the following chapter.

\(^3\)Specifically: continuous feedforward neural networks with at least one internal layer with any continuous sigmoidal nonlinear function as the decision function.
Chapter 5

Fractal analysis

This chapter explains what the fractal dimension is, how it works as a descriptor and how to estimate it.

5.1 Introduction

The fractal dimension can be measured on both topographic maps and common two- and three-dimensional binary images. The fractal dimension of topographic maps has been used as descriptor of paper surface by Chinga in Gary Chinga and Dougherty (2003). The fractal dimension of three-dimensional structures has, to our knowledge, not previously been examined in relation with porous fibre structures. Quantifying fibre structure is one of the main areas of research in the exploration of high-resolution paper images in the paper science domain, c.f. Chapter 4. The fractal dimension is a new tool for quantifying these structures.

The fractal dimension extends the traditional concept of dimension (from now on referred to as ‘topological dimension’) from natural numbers to real numbers. A mathematical definition of the fractal dimension is given in Section 5.2. The term fractal was coined by Mandelbrot (Mandelbrot 1977). Mandelbrot’s theory on fractals is based on the work of the mathematicians Hausdorff and Besicovitch.

The fractal dimension is a scalar between 0 and the topological dimension of the space the set is contained in, or the quantity is not defined (Falconer 2003).

This text will treat the fractal dimension of sets in $\mathbb{N}^2$ and $\mathbb{N}^3$. 
Examples on fractal dimension. For “simple” objects, the fractal dimension equals the topological dimension, e.g. the fractal dimension of a single point is 0, of a line 1, a square 2 and so on. A filled circle in the plane, however, will have a fractal dimension between 1 and 2, it will be closer to 2 than 1 (approximately 1.9), because its structure is more like a square than a line. The border of a circle is structurally close to a line and has a fractal dimension of approximately 1.05.

Another example illustrating how the fractal dimension works is to see how it works in relation to a cube. A perfect equal sided cube will have a measured fractal dimension of 3, independent on how much of the space in the viewing volume it occupies. If we gradually reduce the height of the cube, the fractal dimension will approach 2, as the cube becomes increasingly like a plane. In fact any distortion to a perfect cube will reduce its fractal dimension.

5.2 Mathematical foundation

Fundamental to most definitions of dimension is the idea of ‘measurement at scale $\epsilon$’ (Falconer 2003). The Hausdorff-Besicovitch fractal dimension (from now on called just fractal dimension) is based on this idea. For each $\epsilon$, $\epsilon > 0$, a set is measured in a way that ignores irregularities of size less than $\epsilon$, and how these measurements behave as $\epsilon \to 0$ is observed. A dimension of the set $F$ is then determined by the power law (if any) obeyed by the measure $M_\epsilon(F)$ at scale $\epsilon$ as $\epsilon \to 0$. If constants $c$ and $s$ exists for equation 5.1, $c$ is a scaling constant, it can be said that $F$ has dimension $s$. The fractal dimension is given by this equation, but as an equality.

$$M_\epsilon(F) \sim c\epsilon^{-s} \quad (5.1)$$

The general nature of the fractal dimension does not easily afford computation of the special case when the shape measured is discrete, which is the case we are concerned with (Hjelle 2002).

There are several other dimension measures, which under certain conditions agrees with the Hausdorff dimension. One of these is the boxing-dimension (also known as capacity-dimension and information-dimension). In the case of compact, self-similar fractals, the fractal dimension and the boxing-dimension coincide. There exists several different, equivalent definitions of the boxing-dimension, where equation 5.2 is the most suited for computer approximations (Falconer 2003). In equation 5.2, $d$ is the boxing-dimension, $N(s)$ is the number of boxes of size $s \times s$ that covers the set in question, and $k$ is a constant. The boxing-dimension is defined if and only
if the limit in equation 5.2 exists.

\[ k = \lim_{\epsilon \to 0} \frac{N_\epsilon(s)}{1/\epsilon^d} \]  \hspace{1cm} (5.2)

Equation 5.2 defines the measurements and scale needed to estimate the dimension by equation 5.1, which in the current notation can be written as in equation 5.3 by taking logarithms on both sides. Note that equation 5.3 is an approximation because the limit \( \epsilon \to 0 \) cannot be implemented in a computer because the set \( F \) cannot be measured at sub-pixel size \( (\epsilon < 1) \) when \( F \) is given by a digital image. In fact, it cannot even be measured at \( \epsilon = 1 \) because \( \log(1) = 0 \) and the result of dividing something by 0 is not defined. \( \epsilon = 2 \) is the smallest usable \( \epsilon \) value.

\[ \log(N_\epsilon(s)) = -d \cdot \log(\epsilon) + b \]  \hspace{1cm} (5.3)

This gives that the dimension, \( d \) is the negative of the slope of the line given by calculating equation 5.3 for decreasing sizes of \( \epsilon \). We can then estimate the dimension by the slope of the linear regression line fitting the points given by equation 5.3. The vertical axis is \( \log(N_\epsilon(s)) \) and the horizontal axis is \( -\log(\epsilon) \). The constant \( b \) can safely be ignored because it does not affect the slope, it only translates the line in the vertical direction.

### 5.3 Fractal dimension as a descriptor

The fractal dimension is a descriptive, quantitative measure; it is a statistic, in the sense that it represents an attempt to estimate a single real number for a property of an object (Sarkar and Chaudhuri 1992). The fractal dimension is not a unique, sufficient measure; for example, two objects may appear visually very different from each other another and yet have the same fractal dimension (Smith et al. 1996).

When measuring the fractal dimension of a shape, all of the shape’s features that are smaller than the scale \( \epsilon \) are missed, therefore the results depend not only on the shape itself, but also on the scale used. Fractal dimension is the unique fractional power that yields consistent estimates of a set’s metric properties (Sarkar and Chaudhuri 1992). Because it provides the correct adjustments factor for all those details smaller than the measuring scale, it may also be viewed as a measurement of the shapes’s roughness (Pentland 1984). We have done tests to see how the roughness of a surface and its fractal dimensions coincide. The results are given in Chapter 5.
Smith et al. (1996) and several other studies in biology and medicine that used fractal analysis has shown that changes in the fractal dimension reflect alterations of structural properties according to Sarkar and Chaudhuri (1992).

5.4 Estimating the fractal dimension

When estimating the fractal dimension by equation 5.3, a way of counting boxes must be chosen.

Falconer (2003) lists several equivalent ways of doing the actual box-counting. There are methods based on the number of circles that covers the set $F$, and methods based on coverage of $F$ by squares of side $\epsilon$, as shown in figure 5.1. The location of the squares can either be independent of each other, or the squares can be in a mesh.

![Figure 5.1: Illustration of the box counting principle for two different box sizes](image)

(a) Coverage for box size 64

(b) Coverage for box size 16

Of these, we deem the mesh-variant the most feasible to implement on a computer for calculating the fractal dimension of digital images. The reason for this is that when $F$ is discrete, as it is in digital images, coverage by
5.4. Estimating the fractal dimension

shapes not square (or cubic in the three-dimensional case) will be approximations. Coverage by individual cubes is more complex to calculate, and is problematic due to the rotations of the discrete cubes on the discrete grid that is the image. Additionally, the number of translations when translating individual boxes explodes with decreasing box sizes. Nezádal et al. (2001) states that translating the mesh improves estimates significantly in terms of reproducibility and reliability, and that rotating the mesh will theoretically improve estimates by up to 5%. In cases where $F$ is in a discrete space, as is the case for digital images, the mentioned improvement is not obtainable because rotation of the mesh distorts $F$ when the rotation is different from 90 degrees (see Section 2.1.1).

5.4.1 Selection of parameters

The dimension is estimated by the slope of the regression line given by sampling box count for different box sizes given by equation 5.3. From equation 5.3 it is evident that the points that are the basis for the regression affects the dimension estimate. The start and end point correspond to the start and end box size, respectively. Unfortunately, some points can be far off from the regression line even when counting boxes in a consistent and correct manner. This means that the slope of the regression line can be a bad estimate of dimension in some situations, as illustrated by figure 5.2. Choosing the points the dimension estimate is based on corresponds to choosing box sizes.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure5.2.png}
\caption{A figure illustrating how certain sample points may distort the estimated regression line. The lower line illustrates how an estimate excluding the problem area would be more representative.}
\end{figure}
According to Nezádal et al. (2001), the correct setting of maximum and minimum box sizes is crucial for getting a good estimate. Nezádal et al. (2001) concludes, based on a large number of experiments, that using box sizes larger than $\frac{1}{3}$ of the biggest detail of the fractal will cause improper results, and that using box sizes smaller than 4 does not improve the estimate. Nezádal et al. (2001) suggests a different way of estimating the boxing-dimension because of the problem of choosing box sizes: calculate the box count for a large number of different box sizes, and for each such pair of box count and box size, calculate the dimension according to equation 5.2, disregard the limit, and then choose the most frequent estimate (estimates $E_1$ and $E_2$ are defined as equal if $(E_1 - E_2)^2 < T$, for a constant $T$) as the final estimate. This principle is illustrated in figure 5.3. We have not tested this method due to time constraints.

![Histogram showing the most frequent dimension estimates based on different box sizes](image)

**Figure 5.3:** A histogram showing the most frequent dimension estimates based on different box sizes

### 5.4.2 Algorithms for topographic maps

It is possible to do quick and good estimates of the fractal dimension of grey-level images, interpreted as topographic images (maps). Chen et al. (2003) summarises earlier efforts on estimating the fractal dimension on maps, and suggests two new algorithms for this use. Their improved efficiency and reliability lies in finding the exact number of boxes needed along the z-axis...
based on maximum and minimum value of the z-coordinate of pixels in
columns given by a range of x- and y-coordinates.

The simpler of the two algorithms suggested by Chen, the SDBC algorithm,
is given below:

\[
s = 2 /* s: box size */ \\
s_{\text{max}} = M/2 /* s_{\text{max}}: allowable maximum size of box */
\]

while (s \leq s_{\text{max}}) do
  begin
    r = s/M
    N_{r}(F) = 0
    for each vertical column of boxes do
      begin
        min = minimum grey-level value
        max = maximum grey-level value
        N_{r}(F) = \text{CEIL}( (max-min+1)/s )
      end for
    s = s + 1
  end while

call LINEFIT( log(1/r), log(N_{r}(F)), m, c )

dim(F) = m

CEIL is a function that rounds its argument up to the next integer and
LINEFIT is a function that returns the least squares regression line based
on its first two arguments and returns the values \( m \) and \( c \) from the regression
line given as \( y = mx + c \).

These algorithms do not apply to binary images, because in those types
of images, there is not necessarily a single continuous filled area along the
z-axis for a given x- and y-coordinate, as for maps. As an illustration of
this property, consider the map as a terrain and peaks as mountains. In
a topographic map, which is the terrain seen from above, we cannot see
any caves in the mountains, whether caves exist or not. In one or more
cross-sections of the terrain, though, caves can be seen if they exist.

### 5.4.3 Algorithms for binary images

This algorithm applies to one- to three-dimensional binary images.

The basic algorithm for counting boxes that intersects \( F \) in a mesh is:

for a number of box sizes
  for a number of translations
    minCount = INF
count = 0
for all boxes in mesh
visit box.
if box intersects \( F \) then
    increment count
end for
if count < minCount then minCount = count
end for
boxcount for this size is minCount
end for

Dimension is the slope of the regression line for
the points \(-\log(\text{box size}), \log(\text{boxcount(\text{box size}))}\) for all box sizes.

The basic algorithm has an algorithmic complexity of \( O(B \cdot T \cdot N) \), where \( B \) is the number of different box sizes, \( T \) is the number of translations and \( N \) is the number of pixel sites in the image. This is because complete boxes must be searched in the worst-case to see if \( F \) intersects the box or not.

We have come up with some ideas on how to improve this basic algorithm. These ideas are detailed below.

**Improvement #1: Reducing worst-case occurrences.** When there are large empty regions in the image, a hierarchical storage structure like a quadtree or an octree can be used to speed up calculations of boxcount by avoiding search of regions that are known to be empty. Search of empty regions is the worst-case when visiting boxes to see if the box intersects the object of interest, because all pixels in the box will have to be visited to be sure that the box does not contain the object.

We can use information from the octree (quadtree) to see which regions are worth examining and which are not. To see if a box contains \( F \) we can proceed downwards in the octree (quadtree) stopping when we reach a homogeneous region that completely contains the box either counting the box or not, depending on if the homogeneous region is foreground or background, respectfully. A box can possibly be split over multiple regions in the octree (quadtree). If so, the box should be counted if at least one of these regions is homogeneous and non-background, traversing the tree to the pixel-level if necessary to find out.

**Improvement #2: Doubling of box sizes.** In cases where the box size reduction factor is 2 it is possible to calculate all box counts in a single pass of the image. This is because the boxcount for the smallest box size can be used to calculate the count for the next-smallest and so on in cases where boxes are completely covered by the boxes of next box size.
5.5. Fractal dimension and roughness

Note that all numbers that are divisible by two has the property mentioned above, meaning box count for e.g. box size 5 can be used to calculate the count for 10, and so on, because boxes of size 5 can be completely covered by boxes of size 10. Based on this we see that given some box counts, i.e. the ones with prime sides, we can quickly calculate their two-multiples, e.g. for sizes 2, 3, 5 we can quickly calculate the count for box sizes \[ \{2, 4, 8, 16, 32, \ldots\} \cup \{3, 6, 12, 24, 48, \ldots\} \cup \{5, 10, 20, 40, \ldots\} \].

This improvement can be used simultaneously to improvement #1.

5.4.4 Program code and environment

We have made programs for estimating the fractal dimension on both maps and binary images, see Appendix A.

5.5 Fractal dimension and roughness

We will here present a small test to see whether the fractal dimension of a three-dimensional surface correlates with perceived roughness of that surface.

5.5.1 The test

The surfaces in the test were specified as topographic maps.

A set \( S \) of 100 8-bit images of dimension \( 512 \times 512 \) with intensity value 127 was created. Additive Gaussian noise with a mean of 64 and variance of 25 was added to all images in \( S \). Let’s call the set \( S \) with the added noise for \( S_0 \). Let \( S_i \) be \( S_0 \) with \( i \) repetitive Gaussian blurs with the kernel \( (0.249, 0.707, 1) \). The surface of one image from \( S \) is shown in figure 5.4. The fractal dimension was estimated for all images in the sets \( S_i | i = 0, 1, 2, 5 \) by the plugin for estimating fractal dimension of maps in Appendix A. The parameters to the plugin was set to “automatic box size”, “end box size” 2 and “box reduction factor” 2, with include sub-graph”disabled in all tests.

The results are shown in table 5.1, where \( \bar{d}(i) \) is the average of the dimension estimates of \( S_i \) and ‘Difference’ is the squared difference of the average dimension estimate of \( S_i \) and \( S \), \( (\bar{d}(S) - \bar{d}(S_i))^2 \). The surfaces of \( S_0 \) and the blurred images are shown in figure 5.5.
Table 5.1: Average of dimension estimates.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Dimension estimate</th>
<th>Difference</th>
<th>N-Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d(S)$</td>
<td>2.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$d(S_0)$</td>
<td>2.85</td>
<td>0.72</td>
<td>$1.1e-7$</td>
</tr>
<tr>
<td>$d(S_1)$</td>
<td>2.55</td>
<td>0.30</td>
<td>$4.5e-6$</td>
</tr>
<tr>
<td>$d(S_2)$</td>
<td>2.41</td>
<td>0.17</td>
<td>$1.2e-6$</td>
</tr>
<tr>
<td>$d(S_5)$</td>
<td>2.31</td>
<td>$9.6e-2$</td>
<td>$1.8e-6$</td>
</tr>
</tbody>
</table>

Figure 5.4: Original topographic image (set $S$)
5.5. Fractal dimension and roughness

Figure 5.5: Smoothed surfaces

(a) Added noise (set $S_0$)  (b) Smoothed once (set $S_1$)

(c) Smoothed twice (set $S_2$)  (d) Smoothed five times (set $S_5$)
5.6 Discussion and future work

The fractal dimension (FD) as a descriptor of surface roughness should be compared to other descriptors of surface roughness both in terms of correlation and what they describe. In this respect it might be worthwhile implementing the SBC algorithm of Chen et al. (2003), which produces the best known estimates for the FD of topographic maps.

Additionally, the FD of three-dimensional fibrous structures should be examined to see if it correlates with other parameters, both on the microscopic and macroscopic level. We believe that the FD of the skeleton of a fibrous structure will be a good structural descriptor.

Nezádal et al. (2002) suggests fractal-based methods for evaluating the quality of printed material and printed areas. The method can be used to compare the quality of reproduction of the print points, and seems relevant for studies on reducing the number of missing dots in printouts. It would thus be interesting to use and extended the method presented by this article.
Chapter 6

Conclusion

Absorption mode images were found to be more suited for analyses than phase-contrast images.

Techniques for efficient storage and processing of images were suggested.

The most recent method of binarisation of three-dimensional absorption mode images was explained and commented upon. An improved method of binarisation based on a three-dimensional extension of the Susan principle and automatic threshold detection was suggested.

An overview of the current state of feature extraction from porous materials was presented as well as suggestions for future work in the field. The fractal dimension was judged to be of special interest.

The mathematics behind the fractal dimension and algorithms for determining the fractal dimension of structures were presented.

Programs for three-dimensional Susan filtering and estimating the fractal dimension were developed.
Appendix A

Implementation

We have implemented several plugins for the ImageJ environment. These plugins are freely available on the Internet, at http://www.pvv.org/~perchrh/imagej.

ImageJ is a public domain Java image processing program inspired by NIH Image for the Macintosh. It runs as an online applet and a downloadable application on any computer with a Java 1.1 or later virtual machine. The ImageJ website is http://imagej.sf.net.

A.1 SUSAN plugins

SUSAN means “Smallest Univalence Segment Assimilating Nucleus”. It is a principle introduced by Smith and Brady (1997) that can be used for smoothing (noise-reduction) and edge detection.

A Susan-smoothing plugin has been implemented. Both 2D (D=dimensional) and 3D smoothing are supported. 2D smoothing is also supported on 3D images. A fast-variant of the routine that uses a cubic (or in the 2D case: square) mask of size $3 \times 3 \times 3$ is also supported.

A Susan edge-detection plugin has also been implemented. This plugin has a fast-option similar to the smoothing plugin.

The implementation is based on code from the FSL\textsuperscript{1} project. Their code was made available to ImageJ through the use of the Java Native Interface (JNI). JNI enables the integration of natively compiled code with Java. The native code is placed in a library, and a function is written in Java to

\textsuperscript{1}FMRIB Software Library, written by the Oxford Centre for Functional Magnetic Resonance Imaging of the Brain, and available at http://www.fmrib.ox.ac.uk/fsl/
convert between the different data representations. JNI facilitates this data conversion.

The Susan plugins are under the same license as the FSL, due to the shared code, which is a free non-commercial licence. For details see http://www.fmrib.ox.ac.uk/fsl/fsl/licence.html.

A.2 Plugins for fractal analysis

Plugins for estimating the fractal dimension of 2D and 3D binary images and of 2D grey-level images interpreted at topographic surfaces (maps) has been implemented. They support offsets for added reliability and reproducibility of results. The algorithms used is those presented in Chapter 5.

The plugins are licenced under the GNU public licence (GPL), see http://www.gnu.org/licenses/gpl.html for details.
Bibliography


Kaufman, A.: 1994, Voxels as a computational representation of geometry. URL: [http://citeseer.ist.psu.edu/kaufman94voxels.html](http://citeseer.ist.psu.edu/kaufman94voxels.html)


Index

anisotropic, 7                      N-adjacent, 9
background, 25                     N-connected, 9
binarisation, 13, 19              octree, 13
binary, 7                          paging, 14
bits, 12                           PFI, 1
byte, 12                           pixels, 8

caching, 12                        polygons, 10
calendering, 44                   pore throats, 39
Cropping, 20                      porous phase, 19
ESRF, 4                            ray casting, 11
Fast-Susan, 27                     segmentation, 13
flood-fill algorithm, 22          skeleton, 39
foreground, 25                     structure anisotropy, 35
fractal, 47                        surface fitting, 10
fractal dimension, 47             surface rendering techniques, 9
FSL, 61                            SUSAN, 21

gigabyte (GB), 12                  synchrotron, 3
headbox, 44
information-preserving, 13
iso-valued, 10                     Thresholding, 21
isotropic, 7                       topological dimension, 47
Java, 17                           volume rendering techniques, 10
kilobyte (KB), 12                   voxels, 8
lumen, 40                           X-ray micro tomography, 2
Marching Cubes algorithm, 10       X-ray tomography, 2
medial axis transform, 39          zero order interpolation, 8
megabyte (MB), 12
modulo, 15