NTNU – Trondheim
Norwegian University of
Science and Technology

Development of 3-D Quantitative Analysis of Multi-Photon Microscopy Images

Elisabeth Inge Romijn

Master of Science in Physics and Mathematics
Submission date: June 2012
Supervisor: Magnus Lilledahl, IFY

Norwegian University of Science and Technology
Department of Physics

Abstract

Motivation: Cartilage is a robust but flexible connective tissue found in most joints of the body. The collagen fibres present in the extracellular matrix of cartilage contribute to its tensile strength and stiffness. The purpose of this study is to develop and implement methods to determine the orientation and anisotropy of collagen fibres in 3-D images generated with multi-photon microscopy. The motivation behind developing these techniques is to improve the foundation for further studies on understanding the characteristics of the cartilage matrix. This in turn would give a better foundation for developing artificial matrices and mechanical models, as well as improve diagnostics.

Material and methods: The two methods developed in this study are based on analysing the frequency domain. One is an expansion of a previous developed method by Chaudhuri et al. [1]. This method is based on evaluating the average intensity at different directions in the frequency domain. The direction with the least average intensity is equivalent to the direction of the fibres. The other method is based on thresholding the frequency domain according to intensity followed by fitting an ellipsoid to the remaining data set. The direction of the collagen fibres is equivalent to the direction of the shortest axis of the ellipsoid. These methods are called the sector and ellipsoid method, respectively. To determine how robust these methods are a series of tests were developed. The focus of these tests was to determine if the methods are rotational invariant and if the results are influences by different preprocessing techniques. These preprocessing techniques are: median filtering, deconvolution and skeletonization of the original image containing the collagen fibres. It is also important to determine the sensitivity of the ellipsoid method according to the chosen threshold value. In addition data generated fibres and frequency domains were made to determine the accuracy of the methods.

Results and conclusion: The sector method was not very robust. For most cases there is not one specific direction that has the least average intensity in the frequency domain. Instead there is a quite large minimum area. The ellipsoid method shows promising results. It managed to find the correct direction both for the data generated data sets, but also for the real images. It seems like no preprocessing nor frequency filtering, except for thresholding, is needed to still find the correct direction and its anisotropy. The only remark is that the automatically chosen threshold value was to low for one of the samples. This can probably be improved by making a slight change in the process for choosing a threshold value.

Sammendrag

Motivasjon: Brusk er et robust men fleksibelt bindevev tilstede i de fleste ledd i kroppen. Det er kollagenfibrene i den extracellulærmatrisen til brusk som gir den sin strekkfasthet og stivhet. Målet med dette studiet er å utvikle og implementere metoder for å bestemme retningen til kollagen fiber i 3-D bilder avbildet med multifoton mikroskopi. Motivasjonen bak utviklingen av disse metodene er å forbedre grunnlaget for videre forskning på å forstå egenskapene til brusk. Dette vil igjen gi et bedre fundament for å utvikle kunstig brusk og mekaniske modeller, samt forbedre diagnostikk.

Materialer og metode: De to metodene utviklet i dette studiet er basert på å analysere frekvensdomenet. Et av de er en videreføring av en tideligere utviklet metode laget av Chaudhuri et al.[1]. Denne metoden er basert på å evaluere gjennomsnittsintensiteten av frekvensdomenet i forskjellige retninger. Retningen med minst gjennomsnittsintensitet er ekvivalent med retningen til kollagenfibrene. Den andre metoden er basert på terskling av frekvensdomenet basert på intensitet, fulgt av å tilpasse en ellipsoide til det gjenværende datasettet. Retningen til kollagenfibrene er lik retningen til den korteste aksen av ellipsoiden. Disse metodene vil henholdvis bli kalt sektor- og ellipsoidemetoden. En serie av tester er utviklet og implementert for å bestemme hvor robuste disse metodene er. Målet av disse testene var blant annet å finne ut om disse metodene er rotasjons invariante og om resultatene blir påvirket av forskjellige forbehandlingsteknikker. Disse forbehandlingsteknikkene er et median filter, dekonvolusjon og skeletonization av originalbildet. Det er også viktig å vite hvor avhengig resultatene til ellipsoide metoden er av terskelsverdien. I tillegg har det blitt laget data genererte fibre og tilhørende frekvensdomener for å bestemme presisjonen til metodene.

Resultater og diskusjon: Sektormetoden er ikke veldig robust. For de fleste tilfellene er det ikke kun en retning som har den minste gjennomsnittelige intensiteten i frekvens domenet, men et helt område. Ellipsoidemetoden viser veldig lovende resultater. Den finner riktige retningen både for de data genererte prøvene, men også for de virkelige bildene. Det ser ut til at den trenger verken forbehandling av bildene eller filtrering av frekvensdomenet, bortsett fra terskling, for å komme fram til riktig resultat. Den eneste anmerkningen er at den fant fram til feil intensitet som terskelverdi for en av prøvene. Dette kan sannsynligvis bli forbedret ved en liten endring i prosessen som bestemmer terskelverdien.

Acknowledgements

I would like to thank my supervisor Magnus Lilledahl. His enthusiasm, interest in the project and endless patience is inspiring. In my search for knowledge I spoke with several people within different fields at NTNU. All of these were very open, helpful and showed an interest. I think this is admirable and therefore I would like to thank Morten Kildemo, Paul Anton Letnes, Ingve Simonsen, Richard Blake, Theo Theoharis and John Walmsley.

Contents

Chapter 1

Introduction

Cartilage is a solid, firm, but at the same time flexible connective tissue, and is amongst others present in joints, the rib cage and the nasal cavity. Even though cartilage can handle a considerable amount of pressure, once damaged it has weak ability to heal itself. This can be a problem in cardiothoracic surgery when the hyaline cartilage connected to the ribs has to be cut to enter the chest. It is also related to osteoarthritis, which is a degenerative joint disease associated with ageing and injury of cartilage in joints, often causing chronic pain, deformity of the joint, and deterioration of the cartilage [2, 3].

Hyaline cartilage consists of chondrocytes surrounded by an matrix consisting of collagen fibres, proteoglycan aggregates, multiadhesive glycoproteins and intercellular water. The high amount of water and its mobility in the matrix contributes to the load bearing capacity. It is the collagen fibres that give cartilage its tensile stiffness and strength, and their orientation plays an important part in this. The chondrocyte produce the extracellular matrix. Their low mobility and limited ability to reproduce rapidly is one of the main reasons why cartilage can hardly repair itself [2, 4, 5].

A better knowledge about the characteristics of the extracellular matrix of hyaline cartilage would give a deeper understanding of the cartilage function and would for example give a more solid foundation for making artificial matrices to replace damaged cartilage, develop mechanical models and improve diagnostics.

For this purpose it is important to develop a quantitative method to determine the collagen fibre orientation in cartilage. Some imaging techniques, like second harmonic generation polarimetry [6], magnetic resonance imaging with T_2 relaxation [7] and small angle light scattering [8], can obtain the orientational distribution of the collagen fibres directly. Other imaging techniques produce images of the collagen structure, and thereby require quantitative analytical methods to determine the orientation of collagen fibres in images. Previous work on this topic has already provided several methods. Most of these were developed for 2-D images [1, 9, 10], but also some for 3-D [11], using a variety of techniques, like the use of the Hessian matrix [10], Hough transform [12], gradient vectors [13] and analysis of the frequency domain [1, 9, 14, 15, 16]. Many of these techniques were developed not specifically to determine the collagen fibre orientation in cartilage, but for example in the skin [16], or maybe for a complete different field like the fibre orientation of paper [15].

A variety of imaging techniques can be used to image cartilage. Bulk samples can be imaged by the use of magnetic resonance imaging [7, 17] and X-ray diffraction [18, 19], with the disadvantage of not being able to resolve individual collagen fibres. Imaging techniques with higher resolution but where the tissue has to be sectioned are electron microscopy [20, 21], multi-photon microscopy [10, 22] and polarised light microscopy [23]. The only non-invasive imaging technique mentioned is magnetic resonance imaging. In

this study multi-photon microscopy is used for imaging. The advantages of using multi-photon microscopy are high resolution, no need for sample preparation before imaging and the possibility of 3-D imaging.

The aim of this study is to develop and implement techniques to quantitatively determine the orientation of collagen fibres in 3-D images and degree of anisotropy. These methods are based on analysing the frequency domain. One of these is an expansion of a previous developed method [1, 14, 15], where the intensity distribution of the frequency domain is evaluated at different angles. The other is based on fitting an ellipsoid to the thresholded frequency domain. In addition a series of tests were developed and performed to determine the robustness of these techniques.

Chapter 2

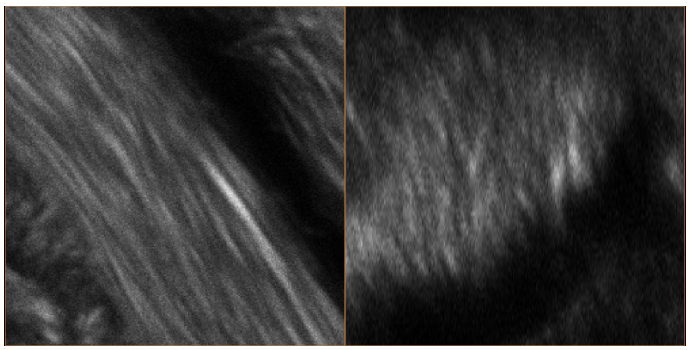
Theory

2.1 Cartilage

Part of the aim of this study is to develop a method to determine collagen orientation in cartilage. Cartilage is a flexible but robust connective tissue. It is an essential part of the human body and is found in most joints. The problem with cartilage is its inability to repair itself [2, 24, 25].

There are three types of cartilage: hyaline, elastic and fibrocartilage. In this study the matrix of hyaline cartilage is examined. The extracellular matrix of hyaline cartilage is a complex tissue with spaces that contain chondrocytes. The matrix itself consists of collagen, proteoglycan aggregates, multiadhesive glycoproteins and intercellular water [2, 24].

The orientation of the collagen fibres in articular cartilage depend on depth from articular surface and there are three distinct layers. These layers are named the superficial, middle and deep layer, where the superficial layer is the closest to the articular surface. In the superficial layer the collagen fibres are oriented parallel to the surface. In the middle layer they are oriented randomly, in coils or aligned in radial bundles. The fibres are oriented normal to the surface in the deepest layer [25, 26]. The chondrocytes within the matrix are shaped in a similar fashion. In the superficial and deep layer they are elongated and respectively oriented parallel and orthogonal to the surface. In the middle layer they have a round appearance [24, 25]. Only samples from the superficial layer are examined in this study. An example is given in figure (2.1)



(a) xy-plane (b) xz-plane

Figure 2.1: Illustrates an image of collagen fibres in hyaline cartilage generated with multi-photon imaging.

2.2 Image properties and imaging

2.2.1 Second harmonic generation microscopy

In this study the multi-photon process called second harmonic generation (SHG) is used to image cartilage. This is a nonlinear photon process whereby two photons of a certain frequency are combined to a photon of twice the energy. It requires an incoming electric field that is strong enough to cause non-symmetric molecules to create an oscillating field at twice the frequency of the incoming field. Within the cartilage matrix the collagen fibres have a strong second-order susceptibility due to their organised, non-centrosymmetric structure [27, 28, 29].

The multi-photon microscope uses a pulsed titanium:sapphire (Ti:sapphire) laser with a centre wavelength of 800 nm. The laser has to have a high flux density in order to induce the SHG. Using a pulsed laser reduces the total power required by the system, as well as reducing the average power applied to the sample [27, 30].

Most of the signal generated in the sample propagates in the same direction as the incoming beam. This part of the signal will always be in phase, since the phase of the molecules in the sample depends on the phase of the exciting beam [27].

SHG only occurs at the focus of the microscope. It is only here the flux density is high enough to induce SHG. Therefore the observed signal must come from the focal spot of the microscope [27, 31]. By moving the sample relative to the focal spot a 3-D image of the sample can be generated.

To filter the excited light from the incoming beam a dichroic beam splitter is used.

An advantage of this imaging technique is that the SHG has no energy dissipation to the sample. This means that the sample receives no energy, as opposed to fluorescence microscopy, and thereby no bleaching of SHG signal [27, 29].

2.2.2 Point spread function

Images created by a microscope is a convolution between the "perfect"image of the object, $I(x, y, z)$, and the point spread function (PSF), $h(x, y, z)$, of the microscope,

$$\begin{aligned} I'(x,y,z) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x+k, y+l, z+m)h(k,l,m)\ dk\ dl\ dm, \\ I' &= I \otimes h. \end{aligned} \tag{2.1}$$

$\otimes$ denotes convolution [30, 32].

The PSF of a microscope describes the smallest object that the microscope can resolve. For a confocal microscope, as opposed to the conventional microscope, the size and shape of the PSF is not only determined by the size and shape of the volume where the signal generation occurs. In addition a pinhole limits the signal to that generated in the focal plane. This is useful for fluorescence microscopy since it removes out-of-focus light. In multi-photon microscopy the signal is already limited in the axial direction, because it is only around the focal spot that the flux density of the beam is high enough for SHG [27, 31]. Accordingly no pinhole is needed and the confocal microscope is turned into a conventional microscope by opening the pinhole fully.

The PSF are of interest is that the effects of the PSF on the image can be partially removed by deconvolution (see section 2.5.1). A theoretical PSF can be calculated based on the properties of the microscope. For a conventional microscope the PSF will be given

by

$$\begin{aligned} h(v,u) &= \left|h'(v,u)\right|^{2p} \\ &= \left|\int_0^1 e^{\frac{iu\rho^2}{2}} J_0(v\rho)\rho \mathrm{d}\rho\right|^{2p}. \end{aligned} \tag{2.2}$$

Here J_0 is a Bessel function of the first kind of the order zero. Depending on whether it is a one, two or three photon process p is either one, two or three. The PSF, h, in this equation is the same as the one in equation (2.1) but given as a function of other coordinates. The connection between the optical radial and axial coordinates, v and u, given in equation (2.2) and the real radial and axial coordinates, r and z, are

$$\begin{aligned} v &= \frac{2\pi}{\lambda} rn \sin\alpha \\ u &= \frac{8\pi}{\lambda} zn \sin^2(\alpha/2). \end{aligned} \tag{2.3}$$

Here $n\sin\alpha$ is the numerical aperture and λ is the incident wavelength.

A theoretical PSF for the microscope can be generated as long as one knows the numerical aperture, the index of refraction of the medium between the objective and its focal point and the incident wavelength [33]. The PSF of the microscope used in this study is illustrated in figure (2.2).

The reason a theoretical PSF is generated instead of measuring the PSF of the microscope is that there are no approximate point sources that can induce SHG.

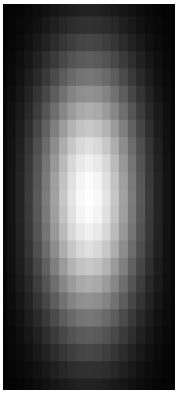


Figure 2.2: Illustrates the xz-plane of a theoretically generated PSF.

2.3 Spatial filters

The data sets used in this study contain noise. This noise might influence the process of determining the orientation of collagen fibres. To remove the noise, different filters can be applied. The ones used in this study are a median and a Savitzky-Golay filter.

2.3.1 Noise reducing median filter

A median filter is a noise reducing, nonlinear filter. Linear filters have the disadvantage of not only removing noise but also reducing image quality and blurring the edges. The median filter on the other hand is good at removing noise whilst mostly preserving the

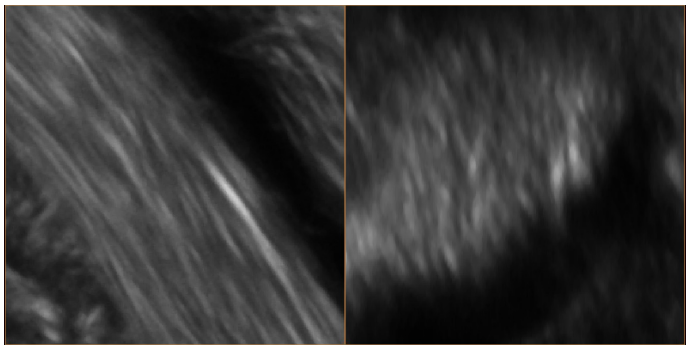
edges. This is due to the fact that every structure smaller than half the filter size is removed whilst the rest is approximately unchanged. It excellently removes salt and pepper and speckle noise, and works better than a Gaussian filter for small levels of Gaussian noise [34].

A median filter replaces the intensity of every voxel in the three-dimensional image by the median of the voxels within a surrounding region, R,

$$I'(x,y,z) = \text{median}\left\{I(x+k, y+l, z+m)\right\}, \quad \text{where } (x+k, y+l, z+m) \in R. \tag{2.4}$$

If the intensities from the voxels within the region R are sorted in ascending order, where I_i has a lower value than I_{i+1}, the median is defined as [35]

$$\text{median}\left\{I_0, I_1, ..., I_n, ..., I_{2n}\right\} = I_n. \tag{2.5}$$



(a) xy-plane (b) xz-plane

Figure 2.3: Illustrates the effect of applying a median filter to the sample from figure (2.1).

2.3.2 Savitzky-Golay filter

The basic idea of the Savitzky-Golay filter is to fit a polynomial to subsections of the image set and replace a chosen voxel within the subsection by the fitted value.

In this section the mathematics of the filter is first explained in one dimension, then expanded to two dimensions. The extension to higher dimensions follows the same pattern.

There are two adjustable parameters in this filter, the window size, m and the order of the polynomial, n. Table (2.1) illustrates a 1-D filter with a window size of nine pixels where the centre pixel will be replaced.

Table 2.1: Example 1-D filter size. The local window coordinates are given by x_i. I_j is the intensity of the pixels within the analysed window.

				x_i				
-4	-3	-2	-1	0	1	2	3	4
I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9

The order of polynomial is based on which will fit the best with the given data set. In general, for intensity plots similar to our own a fourth order polynomial with filter size

similar to the fibres diameter is the best [36]. In this study a fourth order polynomial is used and the appropriate filter size is determined experimentally.

In this example a fourth order polynomial will be used

$$f(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + a_4 x_i^4. \tag{2.6}$$

We want to minimise the error, ϵ, in the following equation

$$f(x_i) = I_i + \epsilon \text{ or } X\vec{a} = \vec{I} + \epsilon. \tag{2.7}$$

Here $\vec{X}_i = [1 \quad x_i \quad x_i^2 \quad x_i^3 \quad x_i^4]$, $X = [\vec{X}_1 \quad \vec{X}_2 \ ... \ \vec{X}_9]^T$, $\vec{a} = [a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4]^T$ and $\vec{I} = [I_1 \quad I_2 \ ... \ I_9]^T$.

To find the optimal coefficients, $\vec{a}$, the least squares fitting is applied

$$\vec{a} = (X^T X)^{-1} X^T \vec{I}. \tag{2.8}$$

Now everything is given to calculate $\vec{a}$. $(X^T X)^{-1} X^T$ is universal for all sections of the image and depends only on how we define the window and which polynomial is fitted. Equation (2.6) shows that it is the easiest to find the approximated intensity of the pixel at the position where $x_i = 0$, since it will be given by a_0. For the given case the first row of $(X^T X)^{-1} X^T$ is illustrated in table (2.2). a_0 can be found by multiplying this row with the image section, $\vec{I}$.

Table 2.2: The first row of $(X^T X)^{-1} X^T$ for the window illustrated in table (2.1) and a polynomial of forth order.

Savitzky-Golay coefficients								
0.0350	-0.1282	0.0699	0.3147	0.4172	0.3147	0.0699	-0.1282	0.0350

By the use of the Savitzky-Golay filter also the first derivative can be found. It will be given by a_1 for the position $x_i = 0$, see equation (2.6). a_1 is found by multiplying the second row of $(X^T X)^{-1} X^T$ with the image section, $\vec{I}$.

At the edges of the image one can not define the window in the same manner, since it will lead to boundary effects. The way we have chosen to solve this problem is by keeping the window size, but moving the position of the pixel that will be replaced. For example if the value of the first pixel in the image has to be fitted the window is defined like the one shown in table (2.3). This will alter the value of $(X^T X)^{-1} X^T$, but the procedure remains the same [36].

Table 2.3: Example of a window defined for the first pixel in an image. The local window coordinates are given by x_i. I_j is the intensity of the pixels within the analysed window.

x_i								
0	1	2	3	4	5	6	7	8
I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9

Applying a 2-D Savitzky-Golay filter is quite similar to applying the 1-D filter.

Table (2.4) illustrates an 5×5 window where the centre pixel is the one that is fitted.

Table 2.4: Example filter size 2-D. The local window coordinates are given by x_i and y_i. I_j is the intensity of the pixels within the analysed window.

		x_i				
		-2	-1	0	1	2
	-2	I_1	I_2	I_3	I_4	I_5
	-1	I_6	I_7	I_8	I_9	I_{10}
y_i	0	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}
	1	I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
	2	I_{21}	I_{22}	I_{23}	I_{24}	I_{25}

To keep things simple a second order polynomial is fitted in the 2D case, given by

$$f(x_i, y_i) = a_{00} + a_{10}x_i + a_{01}y_i + a_{20}x_i^2 + a_{11}x_iy_i + a_{02}y_i^2. \tag{2.9}$$

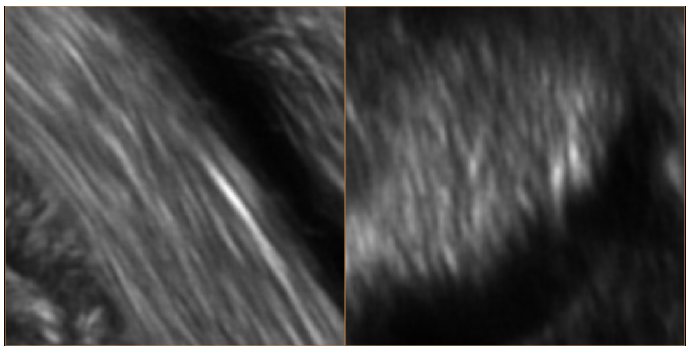
We still want to minimise the error, ϵ, in

$$X\vec{a} = \vec{I} + \epsilon. \tag{2.10}$$

Which is approximated by

$$\vec{a} = (X^TX)^{-1}X^T\vec{I}. \tag{2.11}$$

The rest of the equations just need some minor adjustments. $X_i = [1 \;\; x_i \;\; y_i \;\; x_i^2 \;\; x_iy_i \;\; y_i^2]$, $X = [X_1 \;\; X_2 \;\; ... \;\; X_{25}]^T$, $\vec{a} = [a_{00} \;\; a_{10} \;\; a_{01} \;\; a_{20} \;\; a_{11} \;\; a_{02}]^T$ and $\vec{I} = [I_1 \;\; I_2 \;\; ... \;\; I_{25}]^T$ [37]. In the same manner the 2-D filter can be expanded to a 3-D filter.



(a) xy-plane (b) xz-plane

Figure 2.4: Illustrates the effect of applying a Savitzky-Golay filter to the sample from figure (2.1).

2.4 The frequency domain

2.4.1 Discrete Fourier transform

In this study the frequency domain is analysed to obtain the direction of the collagen fibres.

The discrete Fourier transform (DFT) decomposes a signal into its frequency components. The one dimensional DFT of an Nth-order sequence is given by [38]

$$F(k) = \sum_{x=0}^{N-1} f(x) e^{-\frac{i2\pi k}{N} x}. \tag{2.12}$$

To expand the DFT to three dimensions the fact that it is a separable function is applied. Separability means that a three dimensional DFT can be expressed as the outer product of three one dimensional DFTs [39],

$$\begin{aligned} F(k_x, k_y, k_z) &= \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} \sum_{z=0}^{N_z-1} f(x, y, z) e^{-i2\pi\left(\frac{k_x}{N_x}x + \frac{k_y}{N_y}y + \frac{k_z}{N_z}z\right)} \\ &= \sum_{x=0}^{N_x-1} e^{-\frac{i2\pi k_x}{N_x}x} \sum_{y=0}^{N_y-1} e^{-\frac{i2\pi k_y}{N_y}y} \sum_{z=0}^{N_z-1} e^{-\frac{i2\pi k_z}{N_z}z} f(x, y, z). \end{aligned} \tag{2.13}$$

The computation of the DFT can be very time consuming especially since the size of the image, (N_x, N_y, N_z), often is large. There are techniques to reduce the total number of calculations. The use of symmetry attributes of the DFT for certain size dimensions, for example when N is a power of two, is one of these. The application of these techniques to decrease the computational time is called the fast Fourier transform (FFT) [40].

Spectral leakage is an important source of noise in Fourier images. These are caused by the DFT assuming that the image repeats itself periodically in all directions. The transition from one side of the image to the other might not be gradual. This gives amongst others rise to a strong signal along the main axes in the Fourier space [41].

To reduce this effect a window can be applied. Windowing means making the borders of the image continuous by gradually reducing the intensity of the image to zero at the edges [41]. In this study a Hann window is used for this purpose (see figure (2.5)) [42].

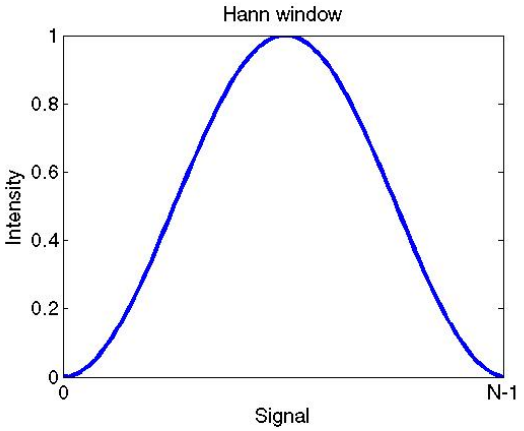


Figure 2.5: Hann window.

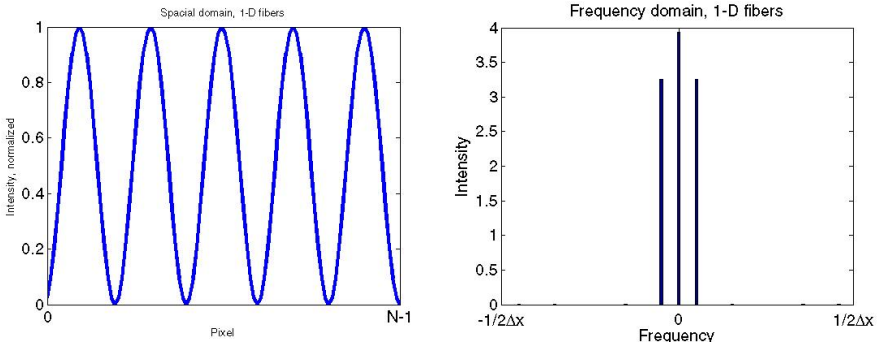
2.4.2 Frequency filtering and interpretation

Fibre orientation in the spatial domain can be determined by analysing the frequency domain. Therefore it is important to understand how structures in the spatial domain affect the frequency domain. To get a deeper understanding of the relation between the spatial domain and the frequency domain some 1-D examples of intensity distributions

and their Fourier transform are examined as well as an expansion to 2-D and 3-D. This is followed by a discussion about the frequency filtering used in this study.

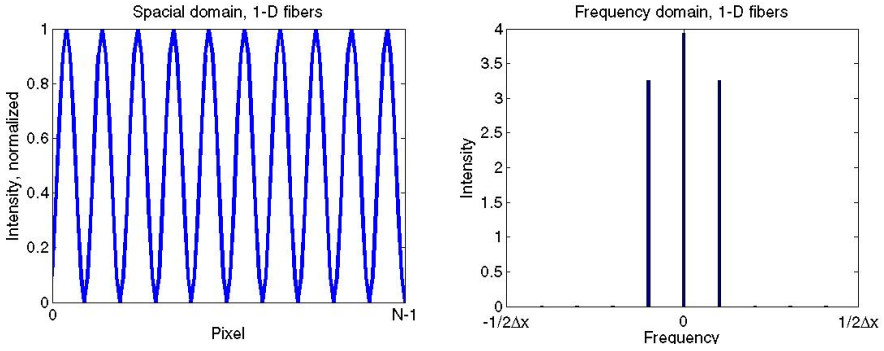
All the computer generated figures in this section are designed in such a way that there is no spectral leakage (section 2.4.1) and therefore no windowing is used.

Lets start by looking at a 1-D line normal to the fibre direction. Assume that the distance between fibres is constant and equal to the diameter of the fibres. The intensity will then have a distribution somewhat like the sinusoidal function illustrated in figure (2.6a). A sine function has a specific wavelength and thereby corresponds to a single frequency, as illustrated in figure (2.6b). The signal in the middle corresponds to the DC component of the image, the frequency domain is symmetric around this point. When the distance between fibres decreases compared to image dimensions, the wavelength becomes shorter. In the frequency domain this action causes the frequency components to move further from the centre (see figure (2.7)).



(a) Spatial domain (b) Frequency domain

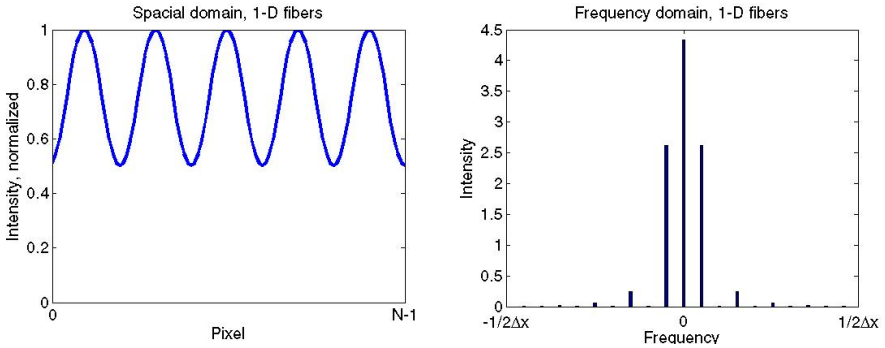
Figure 2.6: The images represent the spatial and frequency domain for 1-D fibres with a diameter equal to the distance between them.



(a) Spatial domain (b) Frequency domain

Figure 2.7: The images represent the spatial and frequency domain for 1-D perfect fibres with a diameter equal to the distance between them. Here the distance between fibres is smaller compared to the image size than in figure (2.6).

What will happen if the fibre diameter is larger than the given distance between them? Physically this is not possible. However blurring of the image increases the apparent fibre thickness, giving rise to the possibility of overlapping intensities. The new intensity distribution has the same wavelength but a decreased amplitude (see figure (2.8a)). As a consequence the same frequencies are present in the frequency domain but with a reduced



(a) Spatial domain (b) Frequency domain

Figure 2.8: The images represent the spatial and frequency domain for 1-D fibres where the distance between fibres is smaller than the diameter of the fibres.

amplitude, as can be seen in figure (2.8b). This can be understood from the 1-D DFT, given by equation (2.12). To explain this lets say that the initial intensity distribution is given by $0.5 + 0.5\sin(x)$ and for the overlapping fibres it is $0.75 + 0.25\sin(x)$. The frequency domain for the original image is

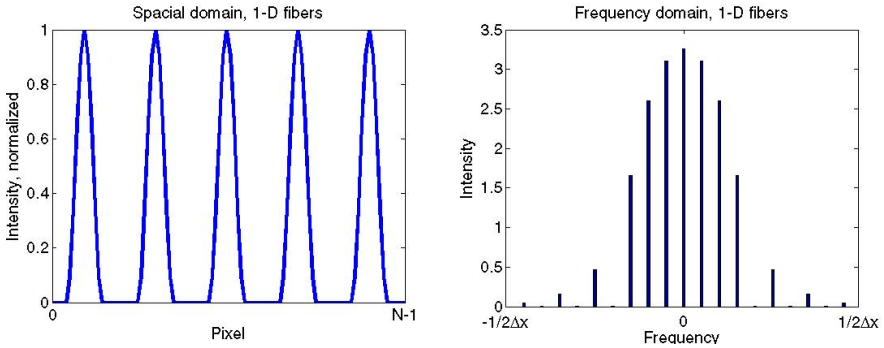
$$\begin{aligned} F(k) &= \sum_{x=0}^{N-1} (0.5 + 0.5\sin(x)) e^{-\frac{i2\pi k}{N}x} \\ &= 0.5 \sum_{x=0}^{N-1} e^{-\frac{i2\pi k}{N}x} + 0.5 \sum_{x=0}^{N-1} \sin(x) e^{-\frac{i2\pi k}{N}x}. \end{aligned} \tag{2.14}$$

The frequency domain for the overlapping fibres will be given by

$$\begin{aligned} F(k) &= \sum_{x=0}^{N-1} (0.75 + 0.25\sin(x)) e^{-\frac{i2\pi k}{N}x} \\ &= 0.75 \sum_{x=0}^{N-1} e^{-\frac{i2\pi k}{N}x} + 0.25 \sum_{x=0}^{N-1} \sin(x) e^{-\frac{i2\pi k}{N}x}. \end{aligned} \tag{2.15}$$

The DC component has increased, whilst the signal in the frequency domain generated by the $\sin(x)$ wave has decreased. Thereby the frequency components of interest have a reduced intensity.

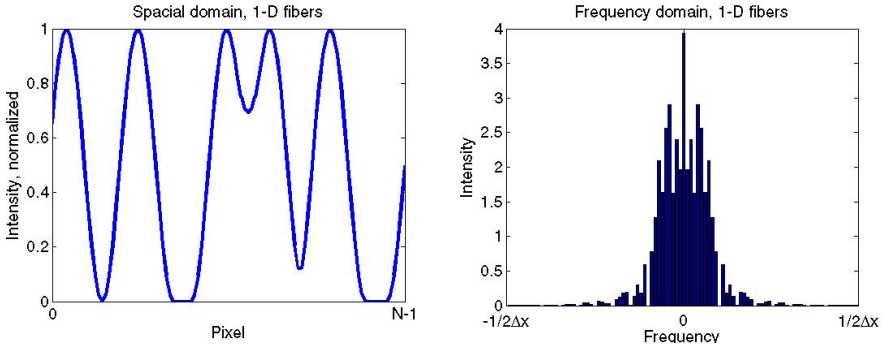
If the diameter is smaller than the distance between fibres the signal becomes more complex. High frequencies become increasingly important as the diameter shrinks. This is because high frequencies are required to constitute the sharp edges in the signal (see figure (2.9)). This causes the frequency domain to spread and high frequencies become more dominant as the diameter of the fibres decreases.



(a) Spatial domain (b) Frequency domain

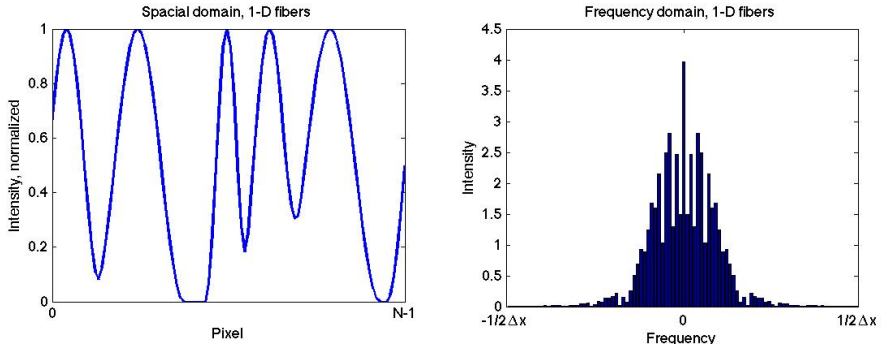
Figure 2.9: The images represent the spatial and frequency domain for 1-D fibres where the distance between fibres is larger than the diameter of the fibres.

In reality the distance between fibres is not constant (see figure (2.10a)). From the previous discussion it is clear that a variety of frequencies are present in this signal (see figure (2.10b)). Even more so if the diameters of the fibres are allowed to vary as well (see figure (2.11)).



(a) Spatial domain (b) Frequency domain

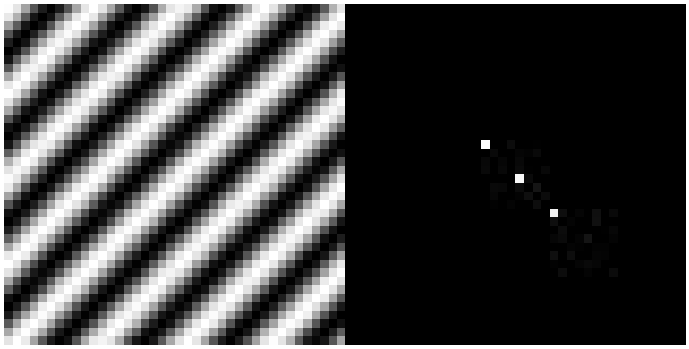
Figure 2.10: The images represent the spatial and frequency domain for 1-D fibres with a variating distance between them.



(a) Spatial domain (b) Frequency domain

Figure 2.11: The images represent the spatial and frequency domain for 1-D fibres with a variating the distance between fibres but also a variating fibre thickness.

Lets return to the neatly organised fibres and add a second dimension. The dimension that is added is the one along the length of the fibres (see figure (2.12a)). If the fibres stretch over the full length of the image, along the fibre direction there will only be a DC component. The direction along the thickness of the fibres on the other hand will be represented by the 1-D signal discussed above. This means that the frequency domain will consist of the 1-D frequency components distributed over a line normal to the fibre direction, as illustrated in figure (2.12b).

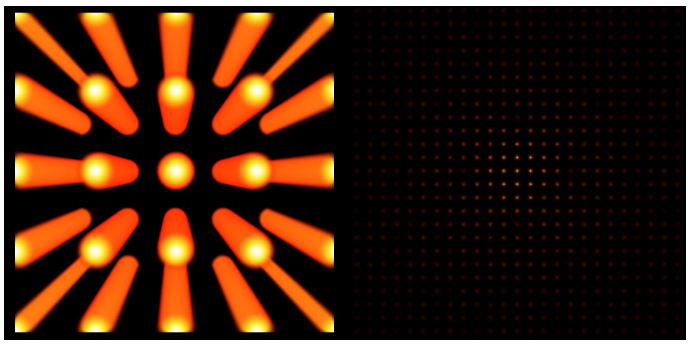


(a) Spatial domain (b) Frequency domain

Figure 2.12: The images represent the spatial and frequency domain for 2-D perfect fibres.

In 3-D, neatly organised fibres, as the one in figure (2.13a), consists of a few specific frequency components in the plane orthogonal to the fibre direction figure (2.13b).

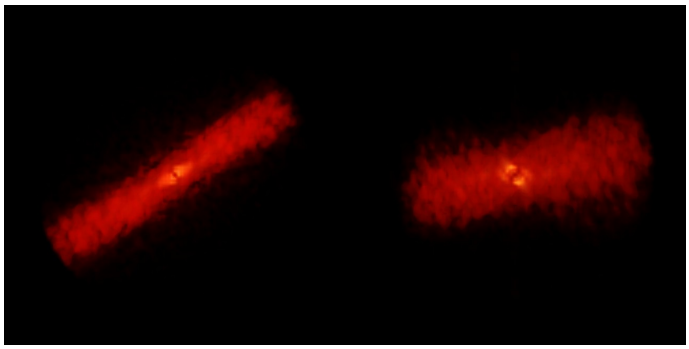
The collagen fibres are not organised as perfect (see figure (2.1)). Similarly to the 1-D case (figure (2.11)) the distance between fibres is not constant, neither is the overall fibre diameter. In addition noise is present in the image. This causes a spread of the frequency components around the dc-component in the plane orthogonal to the fibre direction.



(a) Spatial domain (b) Frequency domain

Figure 2.13: The images represent the spatial and frequency domain for 3-D perfect fibres.

Due to the anisotropy of the PSF (see section 2.2.2) fibres are not round but stretched in the z-direction (see figure (2.1)). In the direction where the fibres are stretched only low frequencies are present, similar to the 1-D case (see figure (2.8)). In the direction where the fibres are not stretched each fibre is distinct, similar to figure (2.9), thereby high frequency components are present in this direction. This causes the frequency components not to be spread like a disc around the dc-component, but like an ellipse (see figure (2.14b)).



(a) xy-plane (b) xz-plane

Figure 2.14: Illustrates the frequency domain of the sample from figure (2.1) with frequency filtering.

Last but not least the fibres do usually not point perfectly in the same direction, there might be some slight changes in intensity along their length, or they might not run through the whole image. This gives rise to frequency components not laying solely in the plane orthogonal to the fibre direction. The more anisotropic the orientation of the fibres are, the more high frequencies will be present in the fibre direction.

Overall this causes the frequency domain to resemble an ellipsoid, where the short axis is in the fibre direction, the medium axis in direction where the diameter of the fibres are stretched, and the long axis is in the direction where the diameter is not stretched (see figure (2.14)). From now on this pattern in the frequency domain will be referred to as an *ellipsoid*.

As the fibre direction becomes more anisotropic the short axis of the ellipsoid will increase. If the fibres are completely randomly organised the frequency domain will be approximately a sphere.

The two methods for determining fibre orientation developed in this study are based on finding the direction of the shortest axis of the ellipsoid in the frequency domain. To make these methods less computationally demanding frequency components that are not of interest are removed from the frequency domain. The highest frequency of interest is the one representing the diameter of the fibres. Thereby all frequency components that are higher can be removed, leaving a spherical data set. The radius of this spherical data set is approximately the same length as the longest axis of the ellipsoid. The same can be done for low frequencies. Low frequencies representing half the image size or more are not of interest. These two filtering processes are a standard in this study. It will be specified if they are not applied.

One of the mentioned techniques involves fitting an ellipsoid to the frequency domain (see section 2.6.3). The problem with this technique is that it only considers the coordinates of each data point and not the intensity. Thus to be able to apply this technique the data points representing the ellipsoid must be extracted. For this purpose segmentation is applied, since the data points belonging to the ellipsoid in general have an higher intensity than the surrounding data points (see figure (2.14)). The appropriate threshold value is approximated by finding the maximal intensity on the surface of the spherical data set. It will be here that the ellipsoid intersects the frequency filtered sphere, and thereby this value will represent the lowest intensity on the ellipsoid.

2.5 Image structure enhancement

As mentioned in section 2.4.2, the stretched shape of the fibres due to the anisotropic PSF causes the ellipsoid in the frequency domain to have a short, medium and long axis instead of resembling a disc. This might cause the medium and short axis to become too similar in length, making it difficult to extract the correct direction of the fibres and it might influence the measurement of the directional anisotropy. As an attempt to reduce/remove the anisotropic effect of the PSF deconvolution and skeletonization are implemented.

2.5.1 Deconvolution

Images created by a microscope are a convolution of a perfect image, I, and the PSF, h,

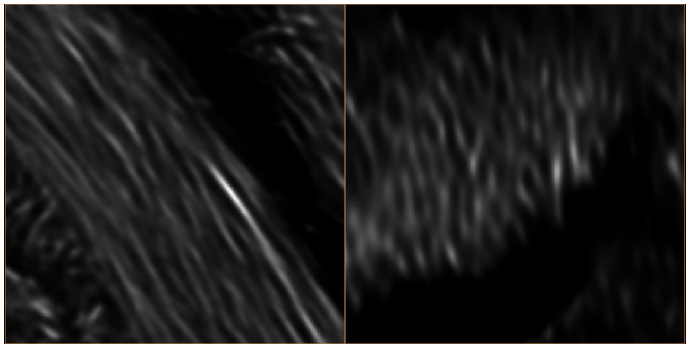
$$I' = I \otimes h. \tag{2.16}$$

Deconvolution, which is the reverse of convolution, can remove the blurring caused by the PSF to a certain degree.

An easy way of applying deconvolution would be by first applying the DFT. A convolution in the spatial domain becomes a multiplication in the frequency domain. Therefore deconvolution becomes a division in the frequency domain,

$$F\{I\} = \frac{F\{I'\}}{F\{h\}}. \tag{2.17}$$

A problem is that there is always noise present in images generated by microscopes. The conversion from photons to electric signal is accompanied by noise, as well as the conversion from an analog to digital signal. In addition there is photon noise, for weak signals the resulting flux might vary due to the quantal nature of light. Noise is amplified by the deconvolution method described above to such a degree that this method gives useless results [30].



(a) xy-plane (b) xz-plane

Figure 2.15: Illustrates deconvolution applied to the sample from figure (2.1).

To overcome this problem more sophisticated deconvolution algorithms have been developed. The one that has been applied in this study is an iterative deconvolution algorithm.

The iteration process works by

1. A start image I_0 is guessed.
2. Convolution of the PSF to the estimate is applied and compared with the image from the microscope I.

3. Based on the error between the last estimate and the image from the microscope a new image is estimated.

4. Any negative voxels are put to zero and any other given restrictions are applied.

5. When the stop criterion is reached, this process is terminated. If not step 2 and onwards is repeated.

Both the start estimate and how the new estimate is generated vary between the different deconvolution techniques.[30, 43]

2.5.2 Skeletonization

Over the years several techniques have been developed to find the skeleton of 3-D images. Many of them are developed for binary images [44, 45, 46, 47], methods have been developed for grayscale images as well [48, 49, 50, 51].

In this study we will expand and implement the 2-D method described by Yu and Bajaj [52] to 3-D.

The basic idea behind this method of skeletonization is to define a vector field of the image, diffuse it in the areas that do not belong to the centre of the fibres, followed by finding the skeleton strength map (SSM) of the remaining vector field.

From the normal gradient field it is not possible to determine where the fibres are located. For this reason the vector field is defined as the vector from each voxel to the nearest neighbour with the lowest intensity. This way all the vectors point away from the centre of the fibres. Mathematically this vector field is defined as

$$g(\vec{r}) = (I(\vec{r}) - I(\vec{r}_{min}))\frac{\vec{r}_{min} - \vec{r}}{\mid \vec{r}_{min} - \vec{r} \mid}. \tag{2.18}$$

Here $I(\vec{r})$ is the intensity of the voxel with location $\vec{r}$ and $\vec{r}_{min}$ is the location of one of the 26 nearest neighbours with the lowest intensity.

The goal of the diffusion is to preserve those vectors that are close to the centre of the fibres whilst smooth those that are not. For this reason anisotropic diffusion, equation (2.19), is applied instead of isotropic diffusion, equation (2.20).

$$\begin{aligned} \frac{du}{dt} &= \nabla \cdot (\mu(\vec{r}, t)\nabla u) \\ \frac{dv}{dt} &= \nabla \cdot (\mu(\vec{r}, t)\nabla v) \\ \frac{dw}{dt} &= \nabla \cdot (\mu(\vec{r}, t)\nabla w) \end{aligned} \tag{2.19}$$

$$\begin{aligned} \frac{du}{dt} &= \mu\nabla^2 u \\ \frac{dv}{dt} &= \mu\nabla^2 v \\ \frac{dw}{dt} &= \mu\nabla^2 w \end{aligned} \tag{2.20}$$

In these equations (u, v, w) are the components of the current vector field. At $t = 0$ these are equal to the components of g, equation (2.18). For isotropic diffusion μ is the diffusion constant, whilst for anisotropic diffusion it determines the degree of diffusion at each location. In the regions where μ is close to one diffusion occurs, whilst where it is close to zero no diffusion occurs [53].

An appropriate function for collagen images would be

$$\mu(\vec{c},\vec{s}) = \begin{cases} e^{K\left(\frac{\vec{c}\cdot\vec{s}}{|\vec{c}||\vec{s}|}-1\right)} & if\ \vec{c}\neq 0\ and\ \vec{s}\neq 0, \\ 0 & if\ \vec{c}=0\ or\ \vec{s}=0. \end{cases} \tag{2.21}$$

Here μ is defined as a function of the central vector, $\vec{c}$, and one of the 6 surrounding vectors, $\vec{s}$. In this study we used the surrounding vector that has the most different direction compared to the central vector. K is an experimentally determined constant, which in this study is set to the value of one. When $\vec{c}$ and $\vec{s}$ point away from each other, as they do near the centre of the fibres, μ is close to zero and thereby little diffusion will occur. If they point in a similar direction, as they will near the edges of the fibres, μ is close to one and diffusion will occur.

Yu and Bajaj [52] do not describe how to rewrite equation (2.20) to its discrete form. Perona and Malik [53] do show this transition in their article. They use a simplified version to reduce the computational time. We choose not to follow their lead, since they use a different weighing function made for finding edges in the image. Rather we rewrite equation (2.19) as direct as possible to a discrete version.

$$\begin{aligned} u_{i,j,k}^{t+1} = u_{i,j,k}^{t} + \lambda[&\mu_{i,j,k}(u_{i+1,j,k}^{t} + u_{i-1,j,k}^{t} + u_{i,j+1,k}^{t} + u_{i,j-1,k}^{t} \\ &+ u_{i,j,k+1}^{t} + u_{i,j,k-1}^{t} - 6u_{i,j,k}^{t}) \\ &+ (\mu_{i+1,j,k} - \mu_{i-1,j,k})(u_{i+1,j,k}^{t} - u_{i-1,j,k}^{t})/\sqrt{2} \\ &+ (\mu_{i,j+1,k} - \mu_{i,j-1,k})(u_{i,j+1,k}^{t} - u_{i,j-1,k}^{t})/\sqrt{2} \\ &+ (\mu_{i,j,k+1} - \mu_{i,j,k-1})(u_{i,j,k+1}^{t} - u_{i,j,k-1}^{t})/\sqrt{2}]. \end{aligned} \tag{2.22}$$

Here $\mu_{i,j,k}$ is the value of equation (2.21) for voxel (i,j,k). According to the Courant-Friedrichs-Lewy step-size restriction $0 \leq \lambda \leq 1/4$ has to be fulfilled for the diffusion to be stable [53, 54]. As a standard in this study λ is set to $1/4$. Equation (2.22) will be similar for for the other two components of the vector field, v and w.

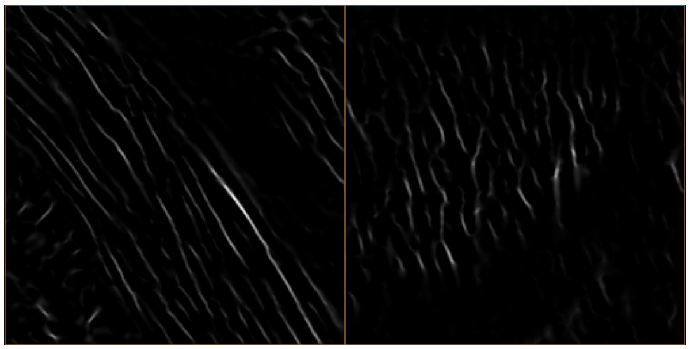
This diffusion method is an iterative process and can therefore be applied several times. Due to the anisotropic property of the diffusion the skeleton will become more distinct when the number of iterations increases, but if taken to far the skeleton will be smoothed as well.

The final step is to find the SSM by using

$$S(\vec{r}) = max\left(0, \sum_{\vec{r}_n \in N(\vec{r})} \frac{g^t(\vec{r}_n)\cdot(\vec{r}_n - \vec{r})}{\mid \vec{r}_n - \vec{r} \mid}\right) \tag{2.23}$$

Here $N(\vec{r})$ is the set of the 26 nearest neighbours to $\vec{r}$.

Yu and Bajaj [52] continue with thresholding the SSM according to Canny's method. Since our purpose is not finding a binary image, but just removing the effects of the anisotropic PSF this last step is not necessary.



(a) xy-plane (b) xz-plane

Figure 2.16: Illustrates skeletonization applied to the sample from figure (2.1).

2.6 Directional analysis

The two methods developed in this study for determining the orientation and anisotropy of the fibres are based on (1) evaluating the intensity within different sectors of the frequency domain (see section 2.6.2), and (2) fitting an ellipsoid to the segmented frequency domain (see section 2.6.3). For further reference they will be denoted the *sector* and *ellipsoid method*, respectivly.

2.6.1 Euler rotation

The sector method (see section 2.6.2) requires rotation of the data set. To do so Euler rotation is applied. Euler's rotation theorem states that any rigid body rotation can be described in terms of three rotations around three axes, not necessarily three different axes. If R_α, R_β and R_γ are the rotational matrices corresponding to the angles α, β and γ, respectively, the full rotation can be described by

$$R = R_\alpha R_\beta R_\gamma. \tag{2.24}$$

It is important to understand that the order of the rotational matrices determines the order of which the rotations are applied. In the example above γ is applied first, followed by β and then α, and is thereby not the same as [55]

$$R = R_\gamma R_\beta R_\alpha. \tag{2.25}$$

In this study the Euler angles that represent rotation around the x, y and z-axis, often called the xyz convention or pitch-roll-yaw, will be used. The rotational matrices around the x, y and z-axis are given by

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{2.26}$$

$$R_\beta = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{2.27}$$

$$R_\gamma = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.28}$$

respectively. All of these rotations are in a counter-clockwise direction [55].

In this study it is particularly useful to know how to rotate the direction given by the spherical coordinates θ and ϕ such that it is oriented along the z-axis. This can be achieved by using the roll, pitch and yaw convention mentioned above. By first rotating an angle $-\phi$ around the z-axis ($\gamma = -\phi$) such that the direction of interest is in the xz-plane. Followed by rotating the new direction an angle of $-\theta$ around the y-axis ($\beta = -\theta$). This gives the rotational matrix

$$\begin{aligned} R &= \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ -\sin\phi & \cos\phi & 0 \\ \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \end{bmatrix}. \end{aligned} \tag{2.29}$$

To examine if this causes the desired rotation lets apply the rotational matrix to the direction $\vec{x} = [\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta]^T$. The result should be the unit vector $\vec{x}' = [0, 0, 1]^T$.

$$\begin{aligned} \vec{x}' &= R\vec{x} \\ &= \begin{bmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ -\sin\phi & \cos\phi & 0 \\ \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \end{bmatrix} \begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix} \\ &= \begin{bmatrix} \sin\theta\cos\theta\cos^2\phi + \sin\theta\cos\theta\sin^2\phi - \sin\theta\cos\theta \\ -\sin\theta\sin\phi\cos\phi + \sin\theta\sin\phi\cos\phi \\ \sin^2\theta\cos^2\phi + \sin^2\theta\sin^2\phi + \cos^2\theta \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{2.30}$$

2.6.2 Sector method

Chaudhuri et al. [1] determined the orientation of fibres in 2-D images by isolating angle bands of the frequency domain. These were transformed to the spatial domain. The directional distribution of the fibres were determined by measuring the abundance of the fibres within one angle band.

One can analyse the frequency domain directly to determine the orientation of the fibres, instead of transforming sections to the spatial domain. This has been done before in 2-D [14, 15]. The method developed and implemented in this study is not only an expansion of these methods to 3-D but is also slightly different in execution.

In short the method used in this study is based on dividing the frequency domain into different angles and evaluating a conic region around each angle to determine in which direction the average intensity is the least. This direction will be the same as the fibre orientation.

The first obstacle is to define angles that are approximately at equal distance from each other. In the θ-direction the distance is defined as $R\mathrm{d}\theta$. In the ϕ-direction the distance should be equal to the one in the θ-direction, giving

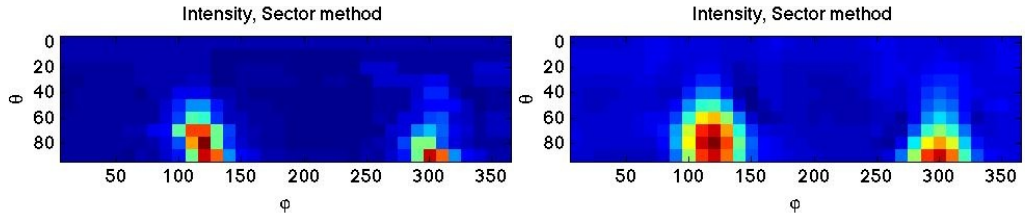
$$R\,\mathrm{d}\theta = R\sin\theta\,\mathrm{d}\phi. \tag{2.31}$$

The value of R is not of interest, since the sectors are cones and thereby the distance from the centre of the image has no meaning. The distance between sectors in the θ-direction

is chosen to be $\mathrm{d}\theta$, meaning that the distance in the ϕ-direction should be $\mathrm{d}\phi{=}\mathrm{d}\theta/\sin\theta$. As a standard in this study a $\mathrm{d}\theta$ of 10° was used.

To evaluate how many data points are in a given direction the whole data set is rotated such that the direction is laying along the z-axis. This is done by the use of Euler rotation (see section 2.6.1). If the data point has an angle θ in the rotated system that is smaller than $\mathrm{d}\theta/\sqrt{2}$ it is within the respective cone and its intensity is added to that direction. The reason the angle $\mathrm{d}\theta/\sqrt{2}$ is used is because $\sqrt{2}\,\mathrm{d}\theta$ is the largest angle between two directions, thereby all data points will at least be within one cone.

The total intensity in one direction is normalised according to the number of data points within the cone. The normalised intensities are plotted in a matrix (see figure (2.17a)). The minimum in this image will determine the orientation of the fibres. To reduce the possibility of an incorrect answer due to noise a Savitzky-Golay filter is applied (figure (2.17b)). There are several ways to define how the anisotropy of the fibres should be calculated. In this study we choose to use the difference between the maximum and minimums value as well as the maximum value of the derivative found with the Savitzky-Golay filter as an indicator. It is important to understand that the difference between the maximum and minimums intensity as well maximum value of the derivative decrease in value as the anisotropy of the fibres increases.



(a) No post processing (b) After Savitzky-Golay filter

Figure 2.17: Intensity distribution of the frequency domain displayed by the use of the sector method.

2.6.3 Least squares ellipsoid specific fitting

This method is based on the method described by Li et al. [56]. In short, the least squares method is applied to find the quadric surface that best matches the data set. To ensure that the quadric surface found is an ellipsoid, ellipsoid specific constraints are added to the least squares fitting.

In general a quadric surface is given by

$$ax^2 + by^2 + cz^2 + 2fyz + 2gxz + 2hxy + 2px + 2qy + 2rz + d = 0. \tag{2.32}$$

We can define I, J, K and L that are invariant under rotation and translation and are given by

$$I = a + b + c \tag{2.33}$$

$$J = ab + bc + ac - f^2 - g^2 - h^2 \tag{2.34}$$

$$K = \begin{vmatrix} a & h & g \\ h & b & f \\ g & f & c \end{vmatrix} \tag{2.35}$$

$$L = \begin{vmatrix} a & h & g & p \\ h & b & f & q \\ g & f & c & r \\ p & q & r & d \end{vmatrix} \tag{2.36}$$

Equation (2.32) represents an ellipsoid when the invariants satisfy: [57]

$$KI > 0,\ J > 0 \text{ and } L < 0. \tag{2.37}$$

There exists a forth inequality that is fulfilled for $k = 4$ if the quadratic surface is indeed an ellipsoid. This inequality is given by

$$kJ - I^2 > 0. \tag{2.38}$$

Here k tells us something about the relation between the shortest and longest axis of the ellipsoid. For $k = 3$ the quadric surface is a sphere. If $k = 4$ the condition guarantees that the surface is an ellipsoid, but it represents only the subset of ellipsoids where the length of the short axis is at least half of the long axis. Generally for any ellipsoid there should exist a $k \geq 4$ that satisfies equation (2.38). The problem is that not all $k > 4$ give solution that is an ellipsoid. So ideally a k should be found that is as large as possible with a solution that still is an ellipsoid.

The goal with this model is to find a quadric equation with the smallest error according to the least squares fitting method whilst fulfilling the inequalities that determines if the solution is an ellipsoid.

First all data points are collected in a matrix D, given by

$$D = [\mathbf{X}_1\ \mathbf{X}_2\ ...\ \mathbf{X}_n], \tag{2.39}$$

where X_i is the vector

$$\mathbf{X}_i = [x_i^2\ y_i^2\ z_i^2\ 2y_iz_i\ 2x_iz_i\ 2x_iy_i\ 2x_i\ 2y_i\ 2z_i\ 1]^T \tag{2.40}$$

and x_i, y_i and z_i are the coordinates of point i. In total there are n datapoints.

To be able to find $\mathbf{v}$, which is given by the constants in equation (2.32)

$$\mathbf{v} = [a\ b\ c\ f\ g\ h\ p\ q\ r\ d]^T, \tag{2.41}$$

the least squares fitting model is applied with the constraint given in equation (2.38),

$$\min \parallel D\mathbf{v} \parallel^2 \quad \text{subject to } kJ - I^2 = 1. \tag{2.42}$$

It would be better if the constraint in equation (2.42) depended on $\mathbf{v}$ instead of J and I. To be able to do so the matrix C_1 is introduced

$$C_1 = \begin{bmatrix} -1 & \frac{k}{2}-1 & \frac{k}{2}-1 & 0 & 0 & 0 \\ \frac{k}{2}-1 & -1 & \frac{k}{2}-1 & 0 & 0 & 0 \\ \frac{k}{2}-1 & \frac{k}{2}-1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k & 0 & 0 \\ 0 & 0 & 0 & 0 & -k & 0 \\ 0 & 0 & 0 & 0 & 0 & -k \end{bmatrix}, \tag{2.43}$$

which is part of a larger matrix

$$C = \begin{bmatrix} C_1 & 0_{6\times4} \\ 0_{4\times6} & 0_{4\times4} \end{bmatrix}. \tag{2.44}$$

Now the constraint can be rewritten to

$$kJ - I^2 = \mathbf{v}^T C \mathbf{v}. \tag{2.45}$$

After this the Lagrange multiplier method can be applied to solve the problem given by equation (2.42). The Lagrange multiplier method is a method for finding the extrema of a function, $f(\mathbf{x})$, which is constrained by the set of equations $g_i(\mathbf{x}) = 0$. The critical points of the function

$$F(\mathbf{x}) = f(\mathbf{x}) - \lambda_i g_i(\mathbf{x}), \tag{2.46}$$

found by solving

$$\frac{\partial F}{\partial x_j} = 0 \text{ and } \frac{\partial F}{\partial \lambda_i} = 0, \tag{2.47}$$

are the extrema of the function $f(\mathbf{x})$ while fulfilling the set of equations $g_i(\mathbf{x}) = 0$. The numbers λ_i are called the Lagrange multipliers [58].

In our case $f(\mathbf{x})$ is $\mathbf{v}^T D D^T \mathbf{v}$, there is only one constraint $\mathbf{v}^T C \mathbf{v} - 1 = 0$ and thereby only one Lagrange multiplier is needed, λ. Equation (2.47) is then given by

$$DD^T \mathbf{v} = \lambda C \mathbf{v} \tag{2.48}$$

$$\mathbf{v}^T C \mathbf{v} = 1. \tag{2.49}$$

Equation (2.48) can be simplified by observing that C consists of large areas with only zeros. By dividing DD^T and $\mathbf{v}$ into subsections given by

$$DD^T = \begin{bmatrix} S_{11} & S_{12} \\ S_{12}^T & S_{22} \end{bmatrix} \quad \text{and} \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}, \tag{2.50}$$

where the size of the matrices S_{11}, S_{12} and S_{22} and vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are given by 6×6, 6×4, 4×4, 6 and 4, respectively, equation (2.48) becomes

$$(S_{11} - \lambda C_1)\mathbf{v}_1 + S_{12}\mathbf{v}_2 = 0 \tag{2.51}$$

$$S_{12}^T \mathbf{v}_1 + S_{22}\mathbf{v}_2 = 0 \tag{2.52}$$

Rearranging equation (2.52), $\mathbf{v}_2$ will be given by

$$\mathbf{v}_2 = -S_{22}^{-1} S_{12}^T \mathbf{v}_1. \tag{2.53}$$

Substituting for $\mathbf{v}_2$ and rewriting equations (2.51) gives

$$C_1^{-1}(S_{11} - S_{12} S_{22}^{-1} S_{12}^T)\mathbf{v}_1 = \lambda \mathbf{v}_1, \tag{2.54}$$

where the solution for $\mathbf{v}_1$ will be given by the eigenvectors of the equation.

This provides a method for finding $\mathbf{v}$ which is the constants of the quadric surface, equation (2.32), but the optimal value for k, which is used to express C_1, is still not found. The optimal value is given by the largest value which still provides an ellipsoid solution for $\mathbf{v}$.

This value is found by an iterative searching process. First k is set to a very large positive number, if the solution for the quadric surface is an ellipsoid according to the inequalities (2.37) the solution is accepted, if not k is divided by half until an ellipsoid is found. It is not certain that this solution is on the boundary of the feasible region. An interval is created where the highest value is set to be twice the value of k, which will be a solution that is not an ellipsoid, and the lowest value is set to k, which is an ellipsoid. Now a new k is approximated to be in the middle of the interval, if this gives a solution

that is an ellipsoid the lowest value of the interval is set to be k, else the highest. This way we are narrowing the interval around the border of the feasible area, until the interval is acceptable small.

Now a quadric surface, which represents an ellipsoid, has been approximated from the data set and the only thing that remains is finding the length and direction of the principle axis. The quadric equation is equal to

$$(\mathbf{X} - \mathbf{X}_0)^T A(\mathbf{X} - \mathbf{X}_0) = 1 \tag{2.55}$$

where A represents the rotation of the ellipsoid and is a positive definite and symmetric matrix, $\mathbf{X}$ is $(x, y, z)^T$ and $\mathbf{X}_0$ is the centre of the ellipsoid. According to the principle axis theorem the eigenvectors of A are the principle directions of the ellipsoid and one divided by the square root of the eigenvalues the corresponding radius of these. For the quadric surface given in equation (2.32) A is [57]

$$A = \begin{bmatrix} a & h & g \\ h & b & f \\ g & f & c \end{bmatrix}. \tag{2.56}$$

In this study this method is used to fit an ellipsoid to the frequency domain (see figure (2.18)). The direction of the shortest axis will be the same as the direction of the collagen fibres. As an indicators of the anisotropy of the image the ratio between the shortest and longest axis is used.

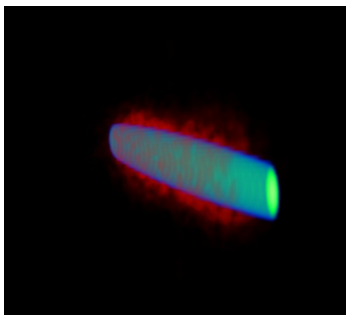


Figure 2.18: The ellipsoid that was fitted to the frequency domain by the use of the ellipsoid method. The frequency domain is in red, the fitted ellipsoid in blue.

Chapter 3

Materials and methods

3.1 Image acquisition

In this study hyaline cartilage from the knee joint of chickens was examined. The cartilage was sliced with a vibratome to samples with a thickness of 100μm and placed directly between the microscope slide and cover slip without fixation. The samples were stored at 4°C.

The equipment used for imaging was a pulsed Ti:sapphire laser (Mira, Coherent) at a wavelength of 800 nm in combination with a confocal microscope (LSM 510 model, Zeiss) with the pinhole fully opened. The laser beam was focused to the sample through an objective with ×40 magnification, water immersion and a numerical aperture of 1.2. The filter used during imaging was a band-pass filter with a range of 385-420nm. The image stacks were generated with different image and voxel sizes, these are specified in section 3.2.

3.2 Image analysis

The image analysis was done by using the computer programs Amira (Visage Imaging, Richmond, Australia) and Matlab (MathWorks, Natick, Massachusetts, U.S.A.). The Matlab code of interest is included in appendix B.

The purpose of this study was to develop and implement methods for determining the orientation of collagen fibres in 3-D images. Basically two methods were developed and their flow chart is illustrated in figure (3.1). Lets start by explaining the different stages in the flow chart and their purpose.

It might be natural to start with the preprocessing, but the purpose of this step is easier to explain at the end. Therefore we will start at step 3, the Fourier transform. Both techniques for determining the direction of the collagen fibres are based on analysing the frequency domain and therefore the Fourier transform of the image has to be taken (see section 2.4.1). This is followed by high and low frequency filtering of the frequency domain, as described in section 2.4.2.

The *sector method* is the left column in the directional analysis. This method produces a plot of the average intensity at different directions in the frequency domain, step 5 (see section 2.6.2). From this plot the direction of the minimum average intensity is extracted. To make this less sensitive to noise a Savitzky-Golay filter is applied to the intensity plot, step 6 (see section 2.3.2).

The *ellipsoid method* is the right column in the directional analysis. The basic idea is to fit an ellipsoid to the frequency domain, as described in section 2.6.3. To do so the

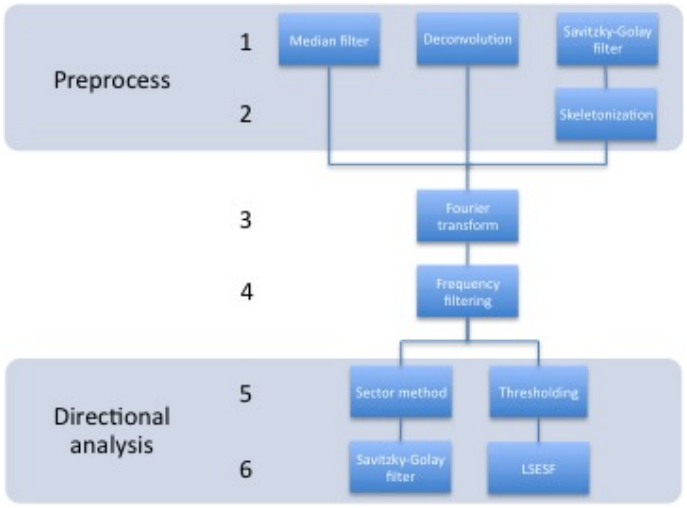


Figure 3.1: Flow chart of the sector and ellipsoid method. The sector method is the left column in the directional analysis. The ellipsoid method is the right. The different preprocessing techniques are optional.

frequency domain has to be thresholded. The determination of the appropriate threshold value is described in section 2.4.2.

As for the preprocessing. The images generated by the microscope are influenced by the anisotropic effect of the PSF (see section 2.2.2) and noise. These will also influence the frequency domain, as explained briefly in section 2.4.2. The median filter, step 1, can remove the noise from the image (see section 2.3.1). Deconvolution, step 1, does not completely remove the effect of the anisotropic PSF but sharpens the image (see section 2.5.1). In this study we wanted to test if the process of skeletonization (see section 2.5.2) could remove the effect of the PSF. The results of this are demonstrated in the first section of the results.

Unless otherwise specified, no preprocessing has been used.

Some steps require input parameters which might vary from study to study. Many of these are already mentioned in chapter 2, to give an overview table (3.1) has been included.

Table 3.1: All the defined values of parameters used in this study.

Process	Step	Parameter	Value	Reference
Savitzky-Golay filter	1	Filter size	$9 \times 9 \times 9$ voxels	Section 2.3.2
		Polynomial	4th order	
	6*	Filter size	9×9 pixels	
		Polynomial	4th order	
Skeletonization	2	K	1	Equation (2.21)
		λ	1/4	Equation (2.22)
		Iterations	10	Section 2.5.2
Sector method	5	$\mathrm{d}\theta$	10°	Section 2.6.2

* The filter is not adapted at the image borders as described in section 2.3.2. The image area comes from a sphere and can thereby continue infinite (see figure (3.2)).

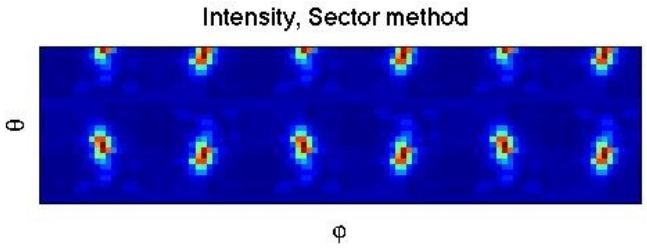
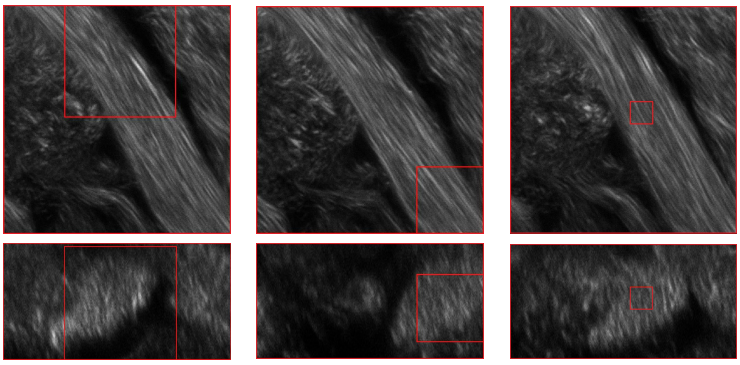


Figure 3.2: This figure illustrates that the intensity plot generated by the sector method (see figure (2.17a)) repeats itself periodically.

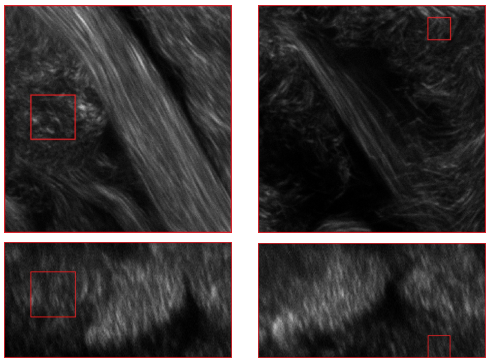
To determine the robustness of the sector and the ellipsoid method a series of tests were developed and executed. These are described in the following sections. For test number 1, 2, 3 and 5, samples from two different data set were used. An overview of the size of the samples and voxel dimensions is given in figure (3.3) and (3.4).



(a) Sample 1 (o), $251 \times 251 \times 251$ voxels

(b) Sample 2 (o), $151 \times 151 \times 151$ voxels

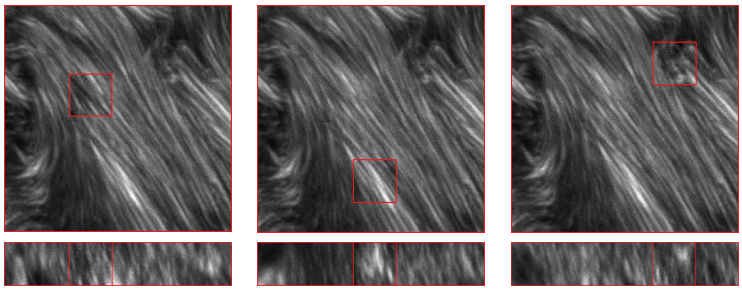
(c) Sample 3 (o), $51 \times 51 \times 51$ voxels



(d) Sample 4 (r), $101 \times 101 \times 101$ voxels

(e) Sample 5 (r), $51 \times 51 \times 51$ voxels

Figure 3.3: Set 1, $73 \times 73 \times 73$ nm voxel dimensions. In each row the upper image is the xy-plane in the middle of the sample stack and the lower image the middle xz-plane. The samples that are orderly arranged are labeled with (o), those that are more randomly arranged with (r). They will be marked similarly in section 4 and appendix A.



(a) Sample 1 (o), $49 \times 49 \times 49$ voxels

(b) Sample 2 (o), $49 \times 49 \times 49$ voxels

(c) Sample 3 (r), $49 \times 49 \times 49$ voxels

Figure 3.4: Set 2, $98 \times 98 \times 98$nm voxel dimensions. The upper image is the xy-plane in the middle of the sample stack and the lower image the middle xz-plane. The samples that are quite orderly arranged are labeled with (o), those that are more randomly arranged with (r). They will be marked similarly in section 4 and appendix A.

3.2.1 Test 1: Rotational invariance

To test if both methods are rotationally invariant the direction and its anisotropy was found after:

- no rotation.
- a 90° anti-clockwise rotation in the xy-plane.
- a 90° clockwise rotation in the xz-plane.

To rotate the samples the rotational matrices described in section 2.6.1 were used.

3.2.2 Test 2: Threshold value dependency

The results generated with the ellipsoid method might be influenced by the thresholding value (step 5 thresholding in figure (3.1)). To investigate how sensitive the results are to the threshold value, plots of the direction and the directional anisotropy versus the threshold value were made.

3.2.3 Test 3: Logarithm versus low frequency filtering

As a standard for all images in this study low frequency filtering is used to remove the low frequencies with a high intensity in the frequency domain (step 4 figure (3.1)). Taking the logarithm of the frequency domain may be an alternative to reduce the relative intensity of the low frequency components. To determine which alternative is better the direction and its anisotropy of the samples were found using both the logarithm and low frequency filtering.

3.2.4 Test 4: Computer generated ellipsoids and fibres

To determine the accuracy of the sector and ellipsoid method, computer generated ellipsoids and fibres were made. These represent the frequency and spatial domain, respectively. A series of different tests, listed below, were executed to explore the limits of both methods.

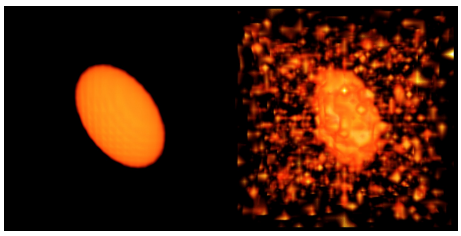
- Discs with increasing thickness: To determine if the measured anisotropy is as expected discs were generated with increasing thickness. Discs are ellipsoids where the two longest axis are the same length, meaning that the fibres in the spatial domain are round (see section 2.4.2). The measured direction and its anisotropy was plotted against the ratio between the shortest and longest axis of the disc. For an example see figure (3.5a).

- Ellipsoids with decreasing difference between the medium and shortest axis: This test was generated to explore if the methods can still find the direction and its anisotropy even if the two longest axes of the ellipsoid are not equal, i.e. if the fibres are not round (see section 2.4.2). The direction and its anisotropy are now plotted against the ratio between the medium and short axis.

- A disc with increasing level of noise: To determine how sensitive both methods are to noise in the frequency domain increased levels of Gaussian noise were added to an image containing a disc. The resulting direction and its anisotropy were plotted against the variance of the Gaussian noise. For an example see figure (3.5b).

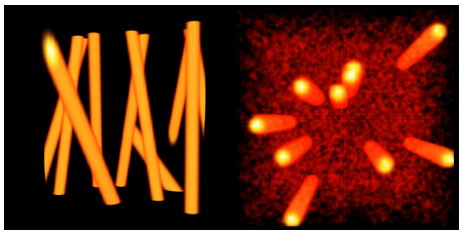
- Round fibres with increasing directional anisotropy: The relation between the spatial domain and the frequency domain is very complicated (see section 2.4.2). The measured direction and its anisotropy were plotted against increasing level of directional anisotropy of the fibres in the spatial domain. This is done to examine if the measured directional anisotropy is as expected and if the general direction can still be determined. To increase the level of directional anisotropy the fibre angle θ was chosen randomly between $0°$ and $n90°$, where n is a number between zero and one, according to the level of anisotropy. For an example see figure (3.6a).

- Fibres with increasing level of noise: To determine if noise in the spatial domain influences the results generated by both methods, increased levels of noise was added to perfectly aligned fibres. The measured direction and its anisotropy was plotted against the variance of the Gaussian noise. For an example see figure (3.6b).



(a) Ellipsoid (b) Ellipsoid with noise

Figure 3.5: Example of computer generated ellipsoids.



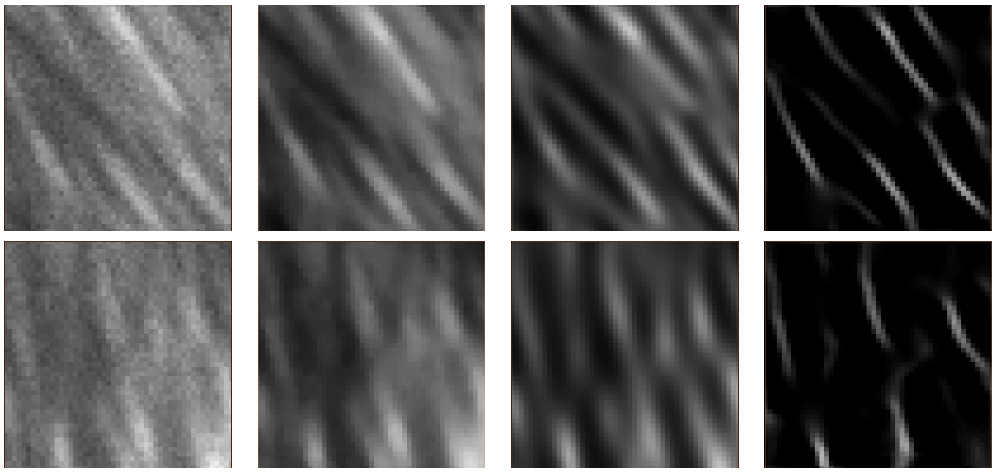
(a) Fibres with high directional anisotropy

(b) Fibres with noise

Figure 3.6: Example of computer generated fibres.

3.2.5 Test 5: Preprocessing techniques

The different preprocessing methods used in this study (preprocessing figure (3.1)) do not remove the anisotropic effect of the PSF and thereby does not make the fibres round (see figure (3.7)). But they do influence the spatial domain, thereby altering the frequency domain and maybe even the measured results. This might influence the process positively by reducing the effects of noise, and increasing the ratio between the medium and short axis in the frequency domain. But it might also cause the relative stretch of the fibres to increase, and thereby decreasing the ratio between the medium and short axis in the frequency domain.



(a) Original

(b) Median filter

(c) Deconvolution

(d) Savitzky-Golay filter and skeletonization

Figure 3.7: Data set 2, sample 1, imaged after different preprocessing techniques. The upper row is a section in the xy-plane, and the lower in the xz-plane.

Chapter 4

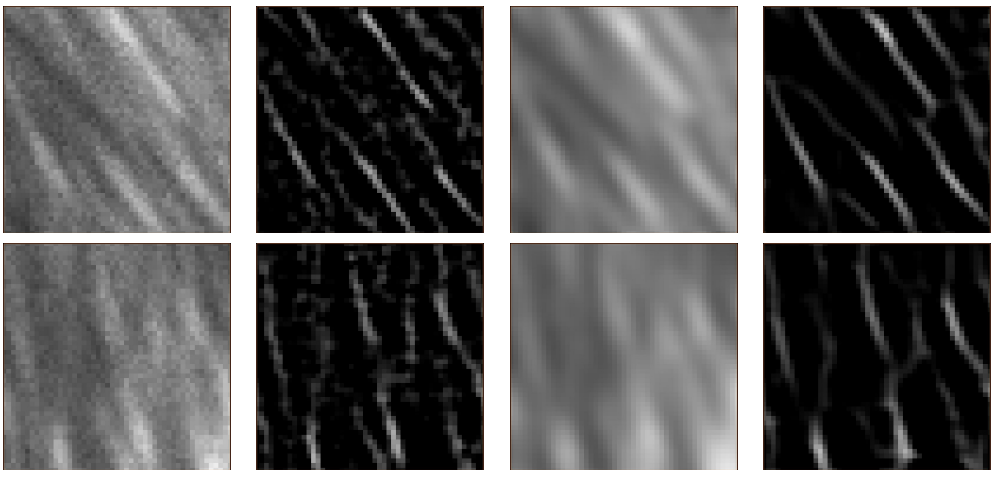
Results

The main results are presented here. The complete results are given in appendix A.

4.1 Skeletonization

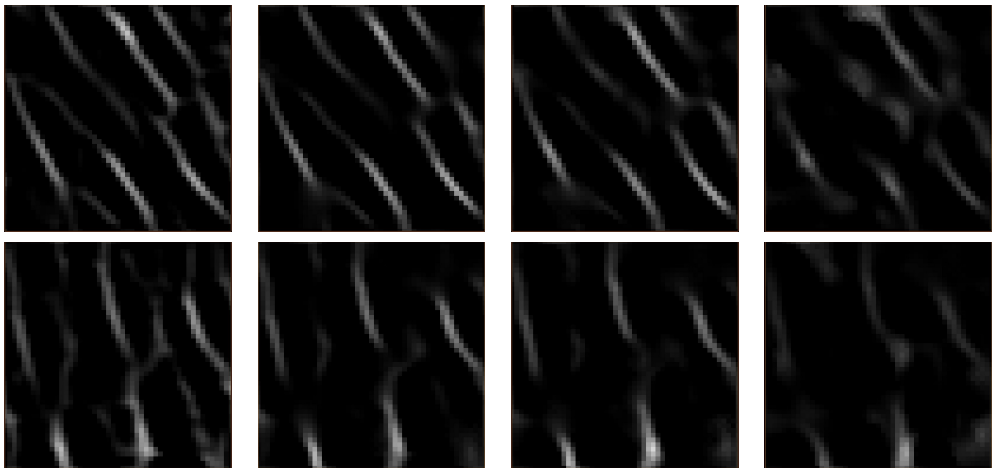
As mentioned in section 3.2 skeletonization is applied to remove the effects of the anisotropic PSF before Fourier transforming (see figure (3.1)). Figure (4.1) illustrates how a Savitzky-Golay filter (step 1 figure (3.1)) influences the results of the skeletonization. The resulting fibres are more smooth and the image contains less noise.

The fibres are not round after one iteration (see figure (4.1d)). As the number of iterations increases the fibres become less stretched (see figure (4.2)). After a certain amount of iterations this effect stagnates and the fibres start to fade (see figure (4.2d)). All in all it looks like the relative stretch of the fibres is larger when skeletonization is applied (see figure (4.2b)) than when nothing is done to the image (see figure (4.1a)).



(a) Original (b) Skeletonization (c) Savitzky-Golay filter (d) Savitzky-Golay filter and skeletonization

Figure 4.1: Data set 2, sample 1. Illustrates the difference in the results for the skeletonization when a Savitzky-Golay filter is applied. Only one iteration is done in the diffusion process. The upper row is a section in the xy-plane, and the lower in the xz-plane.



(a) One iteration (b) Ten iterations (c) Twenty iterations (d) Fourty iterations

Figure 4.2: Data set 2, sample 1. Illustrates how the number of iterations in the diffusion process influence the shape of the fibres. The upper row is a section in the xy-plane, and the lower in the xz-plane.

4.2 Sector and ellipsoid method

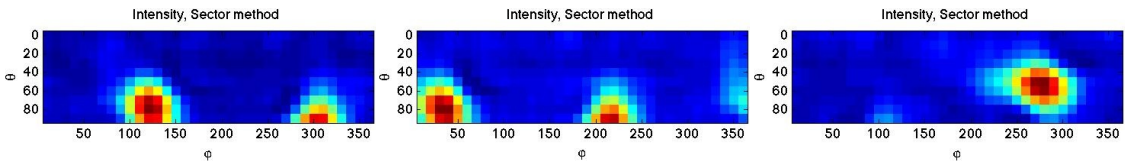
The results of the tests mentioned in section 3.2 are given here.

4.2.1 Test 1: Rotational invariance

The results of the rotational test for the sector method are given in table (4.1). These results are not constant when the data set is submitted to rotation. The anisotropy as defined by the difference between the maximum and minimum average intensity is very high for all samples. It should not be able to have a value larger than one. The Savitzky-Golay filter (step 1 figure (3.1)) can influence the intensity plot to contain negative values, causing the difference to become larger than one.

The results from the ellipsoid test are given in appendix A. These results remain the same when the data set is rotated and thereby illustrate that the ellipsoid method is rotationally invariant.

Figure (4.3) has been included to explore why the sector method is not rotationally invariant. This figure shows the intensity plots generated by the sector method after no rotation and 90° rotation in the xy- and xz-plane. The maximum is very clear in each of these plots, whilst the minimum is more difficult to extract.



(a) No rotation (b) 90° rotation in the xy-plane (c) 90° rotation in the xz-plane

Figure 4.3: Intensity distribution generated with the sector method for data set 2, sample 1 (o).

Table 4.1: Rotational test: results sector method.

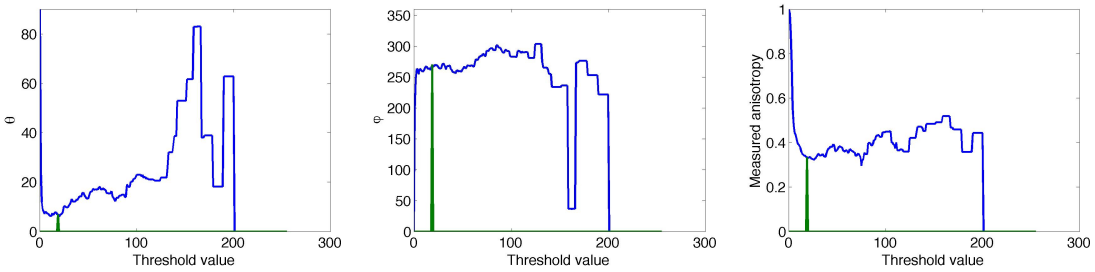
		Data set	1					2		
		Sample	1 (o)	2 (o)	3 (o)	4 (r)	5 (r)	1 (o)	2 (o)	3 (r)
Direction	θ	No rotation	50°	40°	60°	30°	0°	30°	60°	30°
		xy-plane	50°	40°	60°	30°	0°	50°	60°	30°
		xz-plane	60°	67°	64°	31°	22°	67°	80°	36°
	ϕ	No rotation	150°	150°	160°	290°	290°	220°	270°	210°
		xy-plane	150°	150°	150°	290°	200°	160°	160°	220°
		xz-plane	270°	160°	164°	109°	26°	160°	260°	211°
Anisotropy	Difference†	No rotation	0.99	0.99	0.95	0.95	0.97	0.96	1.01	0.96
		xy-plane	1.00	0.98	0.96	0.95	0.98	1.00	1.01	0.95
		xz-plane	0.99	0.99	0.96	0.95	0.93	0.98	0.99	0.93
	Derivative‡	No rotation	0.17	0.17	0.14	0.23	0.19	0.16	0.19	0.15
		xy-plane	0.16	0.17	0.15	0.20	0.19	0.17	0.19	0.15
		xz-plane	0.18	0.20	0.18	0.17	0.15	0.19	0.22	0.16

* LFF stands for low frequency filtering.
† Difference between the maximum and minimum value in the intensity plot generated by the sector method.
‡ The maximum value of the derivative of the intensity plot generated by the sector method.

4.2.2 Test 2: Threshold value dependency

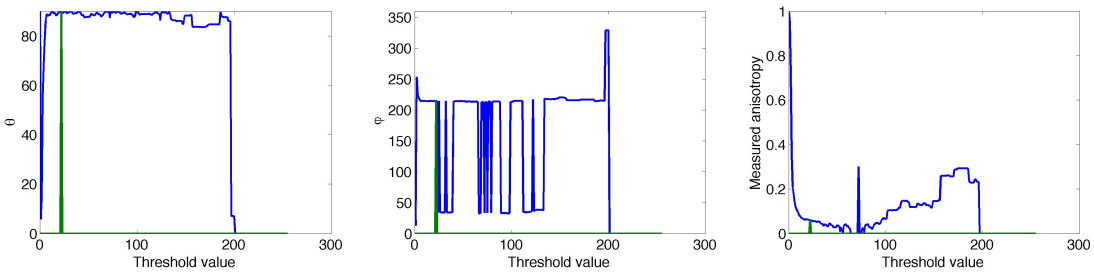
The threshold value dependency was very similar for most samples, and an example of a typical curve is given in figure (4.4). This figure shows a nice and flat area where the results are the same, independent of variation in the threshold value.

Some artifacts were observed and these are illustrated in figure (4.5) and (4.6). The first has similar results for θ and the anisotropy as the other samples, but ϕ is oscillating. The last has also an oscillating ϕ, but in addition the estimated threshold value, the green curve, is in an area where the results change rapidly as a function of the threshold value.



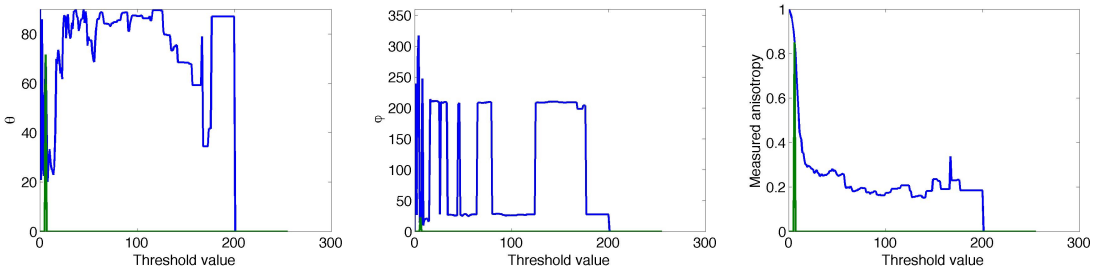
(a) θ (b) ϕ (c) Anisotropy

Figure 4.4: Threshold test: data set 2, sample 3 (r). The green curve shows which threshold value would have been chosen by the program itself.



(a) θ (b) ϕ (c) Anisotropy

Figure 4.5: Threshold test: data set 1, sample 2 (o). The green curve shows which threshold value would have been chosen by the program itself.



(a) θ (b) ϕ (c) Anisotropy

Figure 4.6: Threshold test: data set 1, sample 3 (o). The green curve shows which threshold value would have been chosen by the program itself.

4.2.3 Test 3: Logarithm versus low frequency filtering

The results for the sector method using the logarithm versus low frequency filtering are given in table (4.2). Not only do the results vary between the two, but θ is constant for all samples at a value of 30° when using the logarithm.

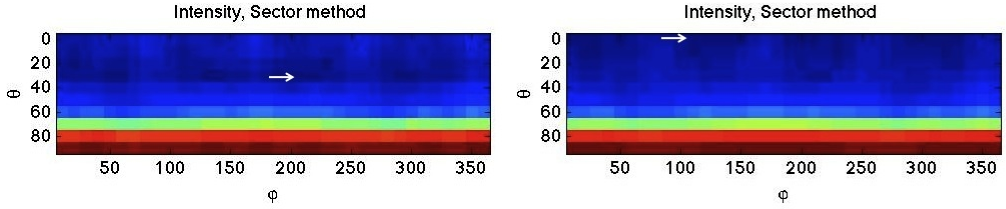
To illustrate the difference between using no filter and a low frequency filter, the intensity distribution of a disc aligned with the z-axis is given in figure (4.7a). In these images there is no noise present to influence the intensity plot and the correct direction is known. As can be seen the minimums intensity point is not at a θ of 0° when no filtering is applied, even though the intensity of the low frequency voxels is not much higher than the rest of the frequency domain.

The direction found with the ellipsoid method is constant and is given in appendix A. The only variation is in the directional anisotropy, which has a difference of 0.01 ± 0.01 between the use of a low frequency filter and the logarithm.

Table 4.2: Logarithm vs low frequency filter: results sector method.

		Data set	1					2		
		Sample	1 (o)	2 (o)	3 (o)	4 (r)	5 (r)	1 (o)	2 (o)	3 (r)
Direction	θ	Logarithm	30°	30°	30°	30°	30°	30°	30°	30°
		LFF*	50°	40°	60°	30°	0°	30°	60°	30°
	ϕ	Logarithm	230°	330°	220°	20°	320°	150°	190°	210°
		LFF	150°	150°	160°	290°	290°	220°	270°	210°
Anisotropy	Difference†	Logarithm	0.26	0.25	0.32	0.28	0.35	0.30	0.30	0.26
		LFF	0.99	0.99	0.95	0.95	0.97	0.96	1.01	0.96
	Derivative‡	Logarithm	0.06	0.06	0.07	0.06	0.06	0.06	0.07	0.05
		LFF	0.17	0.17	0.14	0.23	0.19	0.16	0.19	0.15

* LFF stands for low frequency filtering.
† Difference between the maximum and minimum value in the intensity plot generated by the sector method.
‡ The maximum value of the derivative of the intensity plot generated by the sector method.



(a) No preprocessing

(b) Low frequency filtering

Figure 4.7: Intensity distribution generated with the sector method for a perfect disc without and with low frequency filtering. The short axis of the disc is aligned with the z-axis. The arrow indicates where the minimum is located.

4.2.4 Test 4: Computer generated ellipsoids and fibres

4.2.4.1 Ellipsoids

The results for the computer generated ellipsoid with variable short axis, medium axis and noise are given in figure (4.8), (4.9) and (4.10), respectively.

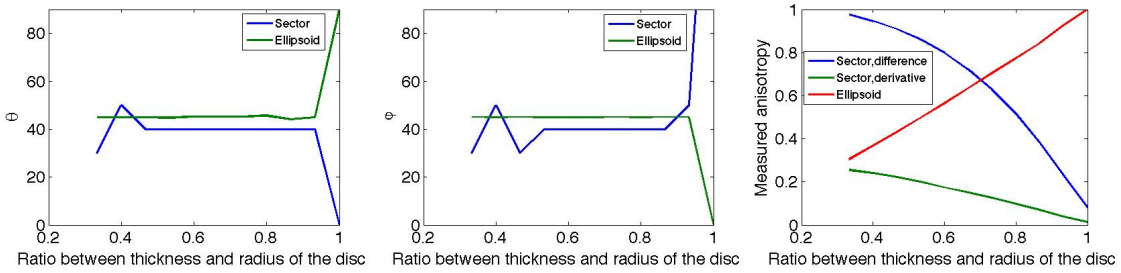
The ellipsoid method reacts as hoped for, i.e. the directional anisotropy increases when short axis increases, whilst the direction remains constant (see figure (4.8)). The measured direction and its anisotropy are not influenced much by the medium axis nor noise (see figure (4.9) and (4.10), respectively).

The sector method has some aberrations in the measured direction when the short axis is increased (see figure (4.8)). The true orientation of the short axis of the disc is $\theta = 45°$ and $\phi = 45°$. So as long as the measured θ and ϕ vary between 40° and 50° the results are approximately correct. For thin discs, i.e. a ratio between the thickness and radius of the disc of 0.47 or less in figure (4.8a) and (4.8b), θ and ϕ are not correct. The measured anisotropy decreases in value, which means that it measures a increase in the

anisotropy of the fibre orientation, as it should be (see figure (4.8c)). To examine why the sector method can not find the correct direction for slim discs the intensity distribution generated with the sector method for a thin and a thicker disc have been included (see figure (4.11)).

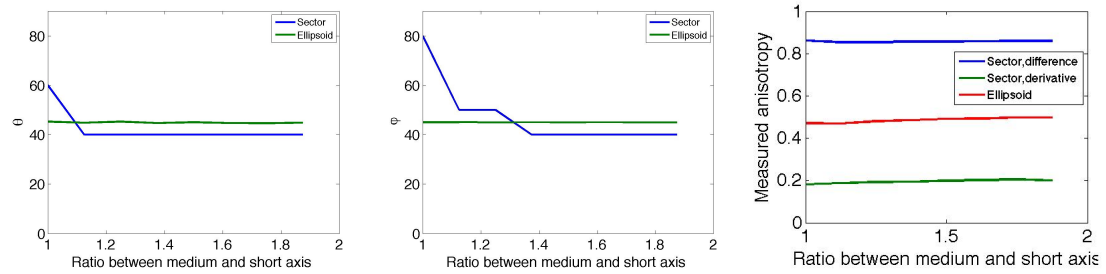
For a variable medium axis of the disc all the results generated with the sector method remain constant (see figure (4.9)), which is positive.

When the frequency domain is subject to noise the measurement of θ remains constant and correct (see figure (4.10a)), but ϕ varies (see figure (4.10b)). The anisotropy measured by both the difference and the derivative decreases as the level of noise increases (see figure (4.10c)).



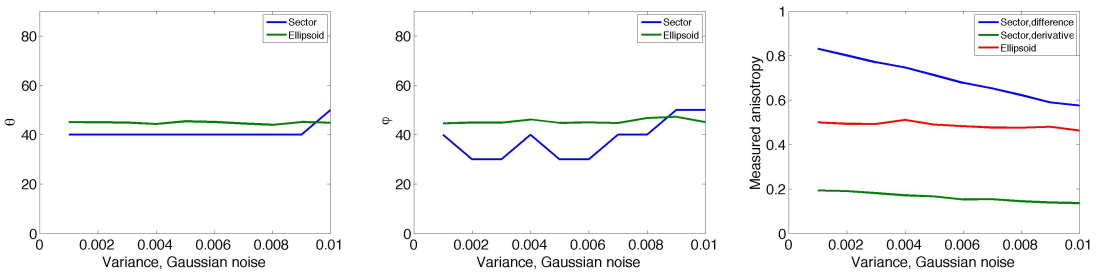
(a) θ (b) ϕ (c) Anisotropy

Figure 4.8: Ellipsoid test, variable short axis. The difference is the difference between the maximum and minimum value in the intensity plot generated by the sector method. The derivative stands for the maximum value of the derivative of the intensity plot generated by the sector method.



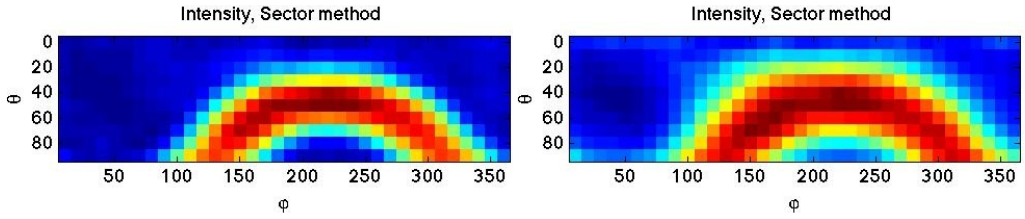
(a) θ (b) ϕ (c) Anisotropy

Figure 4.9: Ellipsoid test, variable medium axis. The difference is the difference between the maximum and minimum value in the intensity plot generated by the sector method. The derivative stands for the maximum value of the derivative of the intensity plot generated by the sector method.



(a) θ (b) ϕ (c) Anisotropy

Figure 4.10: Ellipsoid test, variable noise. The difference is the difference between the maximum and minimum value in the intensity plot generated by the sector method. The derivative stands for the maximum value of the derivative of the intensity plot generated by the sector method.



(a) Small thickness (b) Larger thickness

Figure 4.11: Intensity distribution generated with the sector method for a tilted perfect disc without noise.

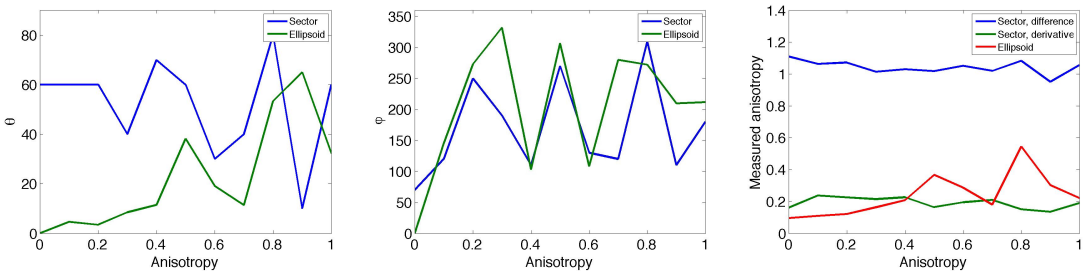
4.2.4.2 Fibres

The results for the computer generated fibres with variable directional anisotropy and noise are given in figure (4.12) and (4.13), respectively.

When increasing the directional anisotropy of the fibres the ellipsoid method has increasing difficulty finding the correct overall direction (see figure (4.12a)). The sector method does not find the correct direction even when the fibres are pointing in exactly the same direction (see figure (4.12a) and (4.12b)). To examine why this is the case figure (4.14) has been included. These images shows that the minimums area in the frequency domain is large, and when applying a Savitzky-Golay filter to the intensity plot the minimum ends up just outside the maximum.

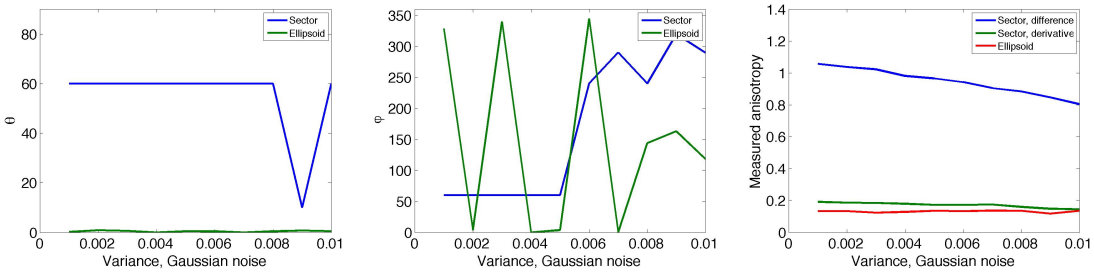
The measured anisotropy should increase in value for the ellipsoid method and decrease for the difference and derivative of the sector method when the directional anisotropy of the fibres increases. In reality they remain fairly constant (see figure (4.12c)). The ellipsoid method has a slight overall increase, but not nearly as much as expected. For complete random fibres the anisotropy should be close to a value of one.

When the level of noise is increased in the spatial domain the direction and its anisotropy measured with the ellipsoid method remains constant (see figure (4.13)). The sector method does not measure the correct direction at all (see figure (4.13a)). The measured anisotropy decreases slightly when using the derivative and a lot when using the difference (see figure (4.13c)).



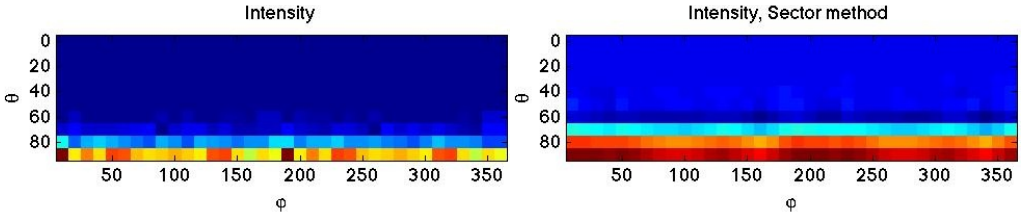
(a) θ (b) ϕ (c) Anisotropy

Figure 4.12: Fibre test, variable anisotropy. The difference is the difference between the maximum and minimum value in the intensity plot generated by the sector method. The derivative stands for the maximum value of the derivative of the intensity plot generated by the sector method.



(a) θ (b) ϕ (c) Anisotropy

Figure 4.13: Fibre test, variable noise. The difference is the difference between the maximum and minimum value in the intensity plot generated by the sector method. The derivative stands for the maximum value of the derivative of the intensity plot generated by the sector method.



(a) Without Savitzky-Golay filter (b) With Savitzky-Golay filter

Figure 4.14: Intensity distribution generated with the sector method of the frequency domain of fibres aligned with the z-axis.

4.2.5 Test 5: Preprocessing techniques

The results for the different preprocessing techniques for the sector and ellipsoid method are given in table (4.3) and (4.4), respectively. For many of the samples the direction measured with the sector method varies (see table (4.3)).

For the ellipsoid method the difference in measured anisotropy between no preprocessing and the median filter, deconvolution and skeletonization is -0.03 ± 0.01, -0.04 ± 0.04 and 0.02 ± 0.10, respectively. The measured anisotropy in general increases for when a median filter and deconvolution is applied. For skeletonization the images that contain fibres that are orderly arranged the measured anisotropy decreases, whilst for those with a random arrangement the measured anisotropy increases.

The measurements of data set 1, sample 3 are left out of the calculations, since the wrong threshold value is estimated by the ellipsoid method (see figure (4.6)).

An example of how the frequency domain is influenced by different preprocessing techniques is given in figure (4.15).

Table 4.3: Preprocessing techniques: results sector method.

		Data set	1					2		
		Sample	1 (o)	2 (o)	3 (o)	4 (r)	5 (r)	1 (o)	2 (o)	3 (r)
Direction	θ	Nothing	50°	40°	60°	30°	0°	30°	60°	30°
		Median filter	50°	40°	0°	0°	30°	30°	60°	30°
		Deconvolution	50°	40°	0°	0°	0°	30°	60°	0°
		Skeletonization	80°	80°	60°	30°	60°	60°	80°	30°
	ϕ	Nothing	150°	150°	160°	290°	290°	220°	270°	210°
		Median filter	150°	150°	200°	20°	220°	220°	270°	210°
		Deconvolution	150°	150°	110°	200°	110°	20°	270°	110°
		Skeletonization	260°	270°	320°	230°	280°	210°	260°	290°
Anisotropy	Difference†	Nothing	0.99	0.99	0.95	0.95	0.97	0.96	1.01	0.96
		Median filter	1.00	1.00	0.98	0.98	0.98	0.96	1.00	0.96
		Deconvolution	1.01	1.02	0.99	0.99	1.00	0.98	1.03	0.99
		Skeletonization	1.02	1.01	0.97	0.95	0.96	0.98	1.06	0.96
	Derivative‡	Nothing	0.17	0.17	0.14	0.23	0.19	0.16	0.19	0.15
		Median filter	0.17	0.17	0.14	0.24	0.18	0.16	0.18	0.15
		Deconvolution	0.17	0.17	0.13	0.24	0.20	0.15	0.17	0.16
		Skeletonization	0.17	0.19	0.17	0.19	0.20	0.17	0.18	0.17

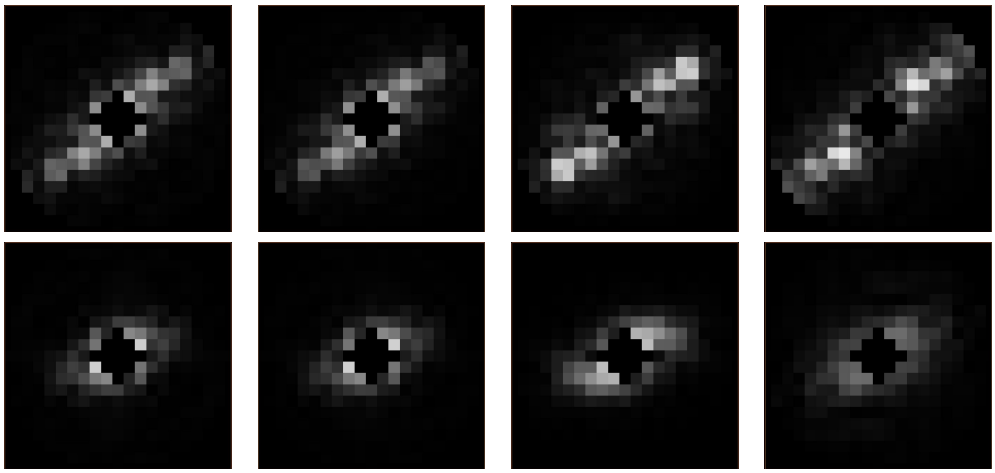
* LFF stands for low frequency filtering.

† Difference between the maximum and minimum value in the intensity plot generated by the sector method.

‡ The maximum value of the derivative of the intensity plot generated by the sector method.

Table 4.4: Preprocessing techniques: results ellipsoid method.

		Data set	1					2		
		Sample	1 (o)	2 (o)	3 (o)	4 (r)	5 (r)	1 (o)	2 (o)	3 (r)
Direction	θ	Nothing	82°	90°	54°	11°	9°	76°	72°	6°
		Median filter	79°	88°	7°	9°	9°	70°	72°	8°
		Deconvolution	69°	84°	4°	9°	10°	62°	71°	7°
		Skeletonization	83°	89°	89°	14°	20°	72°	72°	13°
	ϕ	Nothing	213°	214°	16°	288°	243°	215°	216°	266°
		Median filter	214°	215°	1°	289°	234°	217°	216°	272°
		Deconvolution	216°	214°	290°	286°	252°	217°	216°	294°
		Skeletonization	214°	215°	213°	307°	251°	217°	214°	293°
	Anisotropy	Nothing	0.07	0.06	0.83	0.23	0.37	0.20	0.19	0.33
		Median filter	0.10	0.08	0.55	0.25	0.39	0.22	0.21	0.38
		Deconvolution	0.16	0.11	0.44	0.26	0.46	0.22	0.16	0.37
		Skeletonization	0.00	0.00	0.42	0.31	0.53	0.13	0.08	0.24



(a) Original (b) Median filter (c) Deconvolution (d) Skeletonization

Figure 4.15: Data set 2, sample 1. Illustrates how different preprocessing techniques influence the frequency domain. The lack of intensity in the middle is due to low frequency filtering. The upper row is a section in the xy-plane, and the lower in the xz-plane.

Chapter 5

Discussion

The discussion will start with evaluating the skeletonization technique. Subsequently, the sector and ellipsoid method and the results from the different tests will be discussed individually.

5.1 Skeletonization

The purpose of applying skeletonization (see section 2.5.2) is to remove the anisotropic effect of the PSF, i.e. make the stretched fibres round.

Figure (4.1b) illustrates that the result of skeletonization is very noisy if applied to an unprocessed images. The skeletonization process does not diffuse the image at maximum intensity points in the image. Due to noise in the original image (see figure (4.1a)) these maximum intensity points are not located solely at the centre of the fibres. By applying a noise reduction filter this effect will be removed (see figure (4.1d)). In this study a Savitzky-Golay filter is used for this purpose (see figure (4.1c)).

The Savitzky-Golay filter is not only chosen to remove noise. The intensity within one fibre might vary, especially along the z-direction. This causes the skeletonization to find planes instead of fibres. As an attempt to increase the diffusion in the z-direction the Savitzky-Golay filter is used to remove these intensity fluctuations. It fits a polynomial to the image, thereby smoothing these fluctuations, making the maximum intensity at the centre of the fibres.

The desired result is that after skeletonization the fibres would be round, but unfortunate they become planes (see figure (4.2a)). Increasing the number of iterations helps slightly (see figure (4.2b) and (4.2c)). After a certain number of iterations the fibres start to fade as well (see figure (4.2d)). All in all it may seem like the fibres become sharper in the xy-direction when applying skeletonization, but remain equally stretched in the z-direction.

Yuan et al. [51] developed an algorithm to extract the skeleton of dendrites and spines from 3-D images of neurones made with confocal or multi-photon fluorescent microscopy. Their algorithm is very complex, involving both deconvolution, smoothing using diffusion, gray-scale skeletonization, some algorithms specific for the shape at hand and much more. This indicates that even though skeletonization alone cannot remove the anisotropic effect of the PSF, a combination of several techniques might.

A thing that might improve the results of the skeletonization is using a different weighing function, equation (2.21). The weighing function used in this study is the same as developed by Yu and Bajaj [52]. There might be an equation that is better suited to increases the diffusion in the z-direction more than in the xy-direction.

5.2 Sector method

There are a few things that needs to be pointed out before discussing the results of the tests. Most tests were preformed on images of fibres with no preprocessing. As pointed out in section 2.4.2 the frequency domain will resemble an ellipsoid with a long, medium and short axis instead of a disc, due to that the fibres are oval instead of round. Figure (2.14) is a typical example. Here the long axis is distinct, whilst the two shortest axis are similar in length. This is reflected in the intensity plot generated by the sector method (see figure (2.17a)). The highest intensity is easy to extract, but the minimum intensity is spread over a larger area. The Savitzky-Golay filter is meant to reduce the influence of noise in the intensity plot, but it might influence the location of the minimum wrongfully, as shown in figure (2.17b).

This can influence the results of the tests based on data sets generated with the multi-photon microscope. Test 4 on the other hand is based on data sets containing data generated fibres and ellipsoids, which are not affected by the PSF.

5.2.1 Test 1: Rotational invariance

This test is designed to determine if the sector method is rotational invariant.

Theoretically the sector method (see section 2.6.2) will never be one hundred percent rotational invariant. Firstly the frequency domain can not be divided with an exact equal distance between the cones. Secondly, the frequency domain is a discrete cubic area, therefore each cone will never contain an equal amount of voxels. Thirdly, when making the intensity plot a spherical area is stretched to fit a rectangle, i.e. stretching the area where θ is zero, more than for higher θ-values.

To determine how much this influences the results of the sector method the rotational test is applied (see table (4.1)). Unfortunately the sector method is not rotational invariant for most samples. The intensity plots of a sample where the results were not rotationally invariant is shown in figure (4.3). These figures show a clear maximum area, but the minimum is very hard to extract. Thereby even a small diversity might influence the results.

In other words, if there was a clear minimums point, the sector method might be more robust, but for these samples the sector method is not rotational invariant.

5.2.2 Test 3: Logarithm versus low frequency filtering

In the frequency domain the dc-component and the surrounding low frequencies have a much higher intensity than the rest of the frequency domain (see section 2.4.2). The cones containing these voxels will have a much higher average intensity, causing incorrect results.

To remove this effect two techniques were used. The first is low frequency filtering, which is the standard in this study. Here all frequency components within a radius of 2 voxels from the dc-component are set to zero. The second is taking the logarithm of the frequency domain. The logarithm reduces the difference between low and high intensities, thereby also increasing the effects of noise.

Table (4.2) contains the results for both these techniques. The results when using the logarithm give a constant θ, which is strange.

Low frequency components cover just a few voxels, meaning that not every cone will contain voxels representing these. As illustrated in figure (4.7a), this will influence the intensity plot generated by the sector method even when their intensity is not much higher than the rest of the frequency domain.

For the results generated with the logarithm the effect of the low frequency components is probably not reduced enough. The cones at certain directions where θ is 30° might contain remarkable little low frequency components, thereby having a reduced average intensity.

To conclude low frequency filtering is necessary to remove the effect of low frequency components, since the logarithm is not sufficient.

5.2.3 Test 4: Artificial ellipsoids and fibres

There are three particular properties of the frequency domain that might influence the results of the sector method. These are the length of the shortest axis of the ellipsoid, the length of the medium axis compared to the other two, and the level of noise.

The length of the shortest axis of the ellipsoid, i.e. thickness of the disc if the fibres in the spatial domain are round, will represent the anisotropy of the fibres in the spatial domain. This means that the value of the measured anisotropy should decrease with the thickness, whilst the direction should remain constant. Figure (4.8) illustrates that the anisotropy decreases as it should, but the direction is variable for thin discs. The intensity plot generated with the sector method has a large minimums area for thin discs and decreases as the disc becomes thicker (see figure (4.11)). Meaning that it is more difficult to determine the direction for thin discs compared to thick ones.

The length of the medium axis of the disc will depend on how stretched the fibres are. By varying the length of the medium axis one can examine if the direction can be found even if the fibres are not round. Figure (4.9) shows that the correct direction is found. The anisotropy determined by both the derivative and the difference are not influenced by the length of the medium axis. This means that the sector method can determine the direction and its anisotropy even if the fibres are not completely round.

The frequency domain is quite noisy. To determine how much noise influences the results increased levels of Gaussian noise were added. Noise does both influence the found direction and its anisotropy, (4.10).

The artificial fibres were made to examine how the results were influenced by changes in the spatial domain. First the anisotropy of the fibres were decreased (see figure (4.12)). The general direction is a θ of 0°. For an anisotropy of 0 the fibres are perfectly aligned with the z-axis, but the sector method measures a complete different direction. This is due to the large minimum are in the intensity plot and the Savitzky-Golay filter (see figure (4.14)). As a consequence it is difficult to say anything reasonable about these results, except for that the anisotropy in general does not decrease with decreasing anisotropy. The large minimum averages intensity area makes the sector method sensitive to noise in the spatial domain (see figure (4.13))

5.2.4 Test 5: Preprocessing techniques

We would like to use this section not only to discuss how the different preprocessing techniques influence the results but also if the results could be correct for the given samples.

For the samples that have a low level of anisotropy, i.e. data set 1, sample 1, 2, 3 and data set 2, sample 1 and 2 (see figure (3.3) and (3.4)), the fibres run approximately in a direction of $\theta \approx 90°$ and $\phi \approx 45°$ or $\phi \approx 225°$. The reason it does not look like a ϕ of 45° in figure (3.3) and (3.4) is because the y-axis is zero at the top of the image and increases downwards. The sector method does not find this direction for any of the mentioned samples (see table (4.3)), except for data set 2, sample 2, when skeletonization is applied to the image.

The measured anisotropy defined by the difference between the maximum and minimum value in the intensity plot is for all samples very high (see table (4.3)). Normally it can not become larger than one, but the Savitzky-Golay filter (step 1 figure (3.1)), can influence the image to containe negative intensities. Both the anisotropy defined by the derivative and the difference are not lower for the images with high anisotropy compared to those with a low anisotropy.

To sum up the sector method does not find the correct fibre direction and the anisotropy does not behave as expected, even when different preprocessing techniques are applied.

5.3 Ellipsoid method

The effect of the anisotropic PSF on the data sets might also influence the results of the ellipsoid method. If this is the case, it should appear in test 2, which determines the threshold dependency, since it might cause the direction to become very variable. But also in test 4, where the direction and its anisotropy has to be found for an ellipsoid with a variable medium axis.

5.3.1 Test 1: Rotational invariance

Theoretically the ellipsoid method should be completely rotationally invariant. This is because the threshold value is determined by finding the maximum intensity on the surface of a sphere with a specific radius (see section 2.4.2). Also the fitting of an ellipsoid to the thresholded data set (see section 2.6.3) should be independent of any rigid body rotation.

The results confirm this (see table (A.1)). A thing worth mentioning is the change in ϕ for data set 1, sample 2. At a θ of 90° any angle of $\phi \pm n180°$ ($n = 0, 1, 2, ...$) is the same direction. Thereby it is still the same direction that is found, even though the value of ϕ is not the same.

5.3.2 Test 2: Threshold value dependency

The weakest link of the ellipsoid method is the thresholding of the frequency domain (see section 2.4.2). Including or excluding frequencies on account of intensity might be dubious. This test was made to determine how dependent the results of the ellipsoid method are on the chosen threshold value.

Most samples had results similar to figure (4.4). This means that for a range of threshold values the results remains constant. For these samples the ellipsoid techniques has chosen a threshold value within the constant area. At low threshold values the anisotropy goes towards one, meaning that the data set is spherical. This is because the noise is included to the thresholded data set making it spherical. At high threshold values only a few frequency components remain, which causes the results to jump.

An exception is data set 1, sample 2 (see figure (4.5)). The results for ϕ does not have a constant area, but varies between approximately 30° and 210°. This is due to the same reason as mention in the previous section, i.e. at a θ of 90° any angle of $\phi \pm n180°$ ($n = 0, 1, 2, ...$) is the same direction.

Data set 1, sample 3 (see figure (4.6)) has a distribution similar to set 1, sample 2. The difference is that the chosen threshold values is in an area where still a lot of noise is present, meaning that the program chose a too low threshold value. A value between 40 and 140 would be more appropriate, since the direction and its anisotropy would remain fairly constant.

To conclude, for every sample there is a range of threshold values where the results will remain stable. The challenge is to let the program find a threshold value within this region. The current technique is successful in most cases, but not all. This might be because we chose a too high frequency for the high frequency filtering (see section 2.4.2). Thereby will the longest axis of the ellipsoid be smaller than the radius of the remaining data set, and a too low threshold value is chosen. The frequency used for high frequency filtering in this study is the one corresponding to the diameter of the fibres. It might be better to use the one corresponding to the distance between fibres. This will decrease the radius of the remaining spherical data set and increasing the chosen threshold value.

5.3.3 Test 3: Logarithm versus low frequency filtering

Taking the logarithm of the frequency domain should not influence the results of the ellipsoid method. The logarithm changes the scaling of the intensity distribution, but the same voxel's intensity will determine the threshold value. Thereby the same thresholded data set will be fitted.

The reason there is a difference of 0.01 ± 0.01 between the anisotropy when low frequency filtering is used compared to the logarithm, is that low frequency filtering removes a few voxels in the centre of the data set. The missing voxels would usually draw the fitted ellipsoid a bit more towards the centre of the frequency domain, hence the difference in anisotropy.

This shows that the ellipsoid method actually needs no filtering or other processing of the frequency domain, except for thresholding.

5.3.4 Test 4: Artificial ellipsoids and fibres

The ellipsoid method finds the correct direction of the short axis at different lengths of this axis (see figure (4.8)). The anisotropy increases linearly with decreasing anisotropy as wanted. The results are hardly influenced by the length of the medium axis nor noise (see figure (4.9) and (4.10)). This means that the ellipsoid method can determine the anisotropy and direction of round fibres, but also oval fibres, and is not sensitive to noise in the frequency domain.

The fibre test shows that the ellipsoid method has increasing difficulty to determine the overall direction when the anisotropy increases (see figure (4.12)). This is quite natural because the fibres are allowed to have a more randomly distribution with increasing anisotropy, but it might be that one direction is more represented than others. The anisotropy has a general trend of increasing, but not as much as expected. For fully random orientated fibres the frequency domain should contain a sphere, meaning that the anisotropy should be close to one. The spatial domain only contain ten fibres, and this is probably not enough to cause complete anisotropy.

Noise in the spatial domain does not influence the results of the ellipsoid method (see figure (4.13)). This means that no measurements need to be taken to remove noise from the spatial domain before Fourier transforming.

5.3.5 Test 5: Preprocessing techniques

As mentioned in section 5.2.4 the measured direction for data set 1, sample 1, 2, 3 and data set 2, sample 1 and 2 should be around $\theta \approx 90^\circ$ and $\phi \approx 45^\circ$ or $\phi \approx 225^\circ$. The direction found by the ellipsoid method matches this (see table (4.4)). The exception is data set 1, sample 3, but as mentioned in section 5.3.2 the program had chosen the an incorrect threshold value.

The measured anisotropy acts as expected, i.e. the value is low for the samples with low anisotropy and high for the samples with high anisotropy.

The measured direction is not influenced much by the different preprocessing techniques. The measured anisotropy on the other hand is decreased for the samples with low anisotropy and increased for those with high anisotropy when applying skeletonization. This can be explained by that the fibres become thinner in the xy-plane (see figure (3.7)). Thereby increasing the length of the longest axis in the frequency domain (see figure (4.15)). This reduces the value of the measured anisotropy, but images where the fibres are randomly arranged the frequency domain is less affected.

The question is if the increased difference in anisotropy is worth the extra computational time accompanying skeletonization. To give an example, the largest sample used in this study has an image size of $250 \times 250 \times 250$ voxels. It took more than 12 hours to find its skeleton. On the other hand this might be reduced by simplifying equation (2.22), similarly to Perona and Malik [53].

5.4 Comparison between the sector, ellipsoid and previous developed methods

The sector method has proven not to be very robust. This is a method that is an expansion from a previous developed technique by Chaudhuri et al. [1]. In two dimensions one can find the maximum intensity direction instead of the minimum and just add 90°. In 3-D it is not possible to use the maximum to determine the direction. As illustrated in figure (2.17) and (4.3) the maximum intensity is more distinct than the minimum. If we should guess, it is the same case in 2-D and therefore has the maximum been used to determine the direction. The Savitzky-Golay filter seems to influence the intensity plot wrongfully. When the minimum average intensity area is large, the Savitzky-Golay filter seems to place the minimum directly next to the maximum (see figure (2.17) and (4.14)). Therefore replacing this filter by an other might improve the results of the sector method.

The ellipsoid method on the other hand is working exactly as hoped for. We have not seen any previous work on analysing the 3-D frequency domain nor fitting an ellipsoid to the frequency domain.

An easier way of finding the principal directions of the frequency domain might be the method developed by Denslow et at. [9]. In their study they used a variance-covariance matrix of the intensity spectrum of the frequency domain to determine the direction and its anisotropy of fibres in images. In the mentioned article it is used for 2-D images. This method needs neither thresholding nor fitting, but we do not know how easy it would be to expand this technique to 3-D.

Chapter 6

Conclusion

In this study we have developed and implemented two methods to determine the direction and its anisotropy of collagen fibres in 3-D images generated with multi-photon microscopy. These are the sector and ellipsoid method.

Through a series of tests it was shown that the sector method was not very robust. This is because the frequency domain does not contain one specific direction where the average intensity is the least, but rather a quite large area. The maximum average direction is limited to a much smaller area and is therefore much easier to extract. This is due to the anisotropic effect of the PSF on the spatial domain. Even if the anisotropic effect of the PSF is removed from the images before Fourier transforming, the minimum average intensity might still not be at just one specific direction. This method was originally made for 2-D, and the maximum average intensity was used to determine the direction of the fibres [14]. Therefore is the method probably not working in 3-D. Changes that might improve the results are plotting the average intensity distribution on a hemisphere instead of stretching it to the shape of a rectangular (see figure (2.17)). Also it seems like the Savitzky-Golay filter (step 6 figure (3.1)) influences the results wrongfully, therefore using an other noise reduction filter might improve the results greatly.

The ellipsoid method showed very promising results. It was hardly effected by noise in the frequency domain nor in the spatial domain. It managed to find the correct direction for the data generated samples as well as what we assume is the correct direction for the real samples. There seems to be no need to remove the anisotropic effect of the PSF, since the ellipsoid method manages to find the correct direction and anisotropy. The process of choosing the threshold value did not work for all samples, but with a slight change this problem might be solved. For future research we recommend to test this method on a larger range of samples to make sure this method is robust.

Bibliography

[1] S. Chaudhuri, H. Nguyen, R. M. Rangayyan, S. Walsh, and C. B. Frank, "A fourier domain directional filtering method for analysis of collagen alignment in ligaments," *IEEE Transactions on Biomedical Engineering*, 1987.

[2] M. H. Ross and W. Pawlina, *Histology: a text and atlas*. Lippincott Williams & Wilkins, 2006.

[3] R. J. Williams, *Cartilage Repair Strategies*. Humana Press, 2007.

[4] R. Shirazi and A. Shirazi-Adl, "Deep vertical collagen fibrils play a significant role in mechanics of articular cartilage," *Journal of Orthopaedic Research*, 2008.

[5] D. L. Skaggs, W. H. Warden, and V. C. Mow, "Radial tie fibers influence the tensile properties of the bovine medial meniscus," *Journal of Orthopaedic Research*, 1994.

[6] T. Yasui, Y. Tohno, and T. Araki, "Determination of collagen fiber orientation in human tissue by use of polarization measurement of molecular second-harmonic-generation light," *Applied Optics*, 2004.

[7] M. T. Nieminen, J. Rieppo, J. Töyräs, J. M. Hakumäki, J. Silvennoinen, M. M. Hyttinen, H. J. Helminen, and J. S. Jurvelin, "T2 relaxation reveals spatial collagen architecture in articular cartilage: A comparative quantitative mri and polarized light microscopic study," *Magnetic Resonance in Medicine*, 2001.

[8] P. L. Kronick and M. S. Sacks, "Quantification of vertical-fiber defect in cattle hide by small-angle light scattering," *Connective Tissue Research*, 1991.

[9] S. Denslow, Z. Zhang, R. P. Thompson, and C. F. Lam, "Statistically characterized features for directionality quantitation in patterns and textures," *Pattern Recognition*, 1993.

[10] M. P. Rubbens, A. Driessen-Mol, R. A. Boerboom, M. M. Koppert, H. C. van Assen, B. M. TerHaar Romeny, F. P. T. Baaijens, and C. V. C. Bouten, "Quantification of the temporal evolution of collagen orientation in mechanically conditioned engineered cardiovascular tissues," *Annals of Biomedical Engineering*, 2009.

[11] H. J. Noordmans and A. W. M. Smeulders, "High accuracy tracking of 2d/3d curved line-structures by consecutive cross-section matching," *Pattern Recognition Letters*, 1998.

[12] W. J. Karlon, J. W. Covell, A. D. McCulloch, J. J. Hunter, and O. J. H., "Automated measurement of myofiber disarray in transgenic mice with ventricular expression of ras," *The Anatomical Record*, 1998.

[13] F. Gadala-Maria and F. Parsi, "Measurement of fiber orientation in short-fiber composites using digital image processing," *Polymer Composites*, 1993.

[14] W. M. Petroll, H. D. Cavanagh, P. Barry, P. Andrews, and J. V. Jester, "Quantitative analysis of stress fiber orientation during coreal wound contraction," *Journal of Cell Science*, 1993.

[15] C. F. Yang, C. M. Crosby, A. R. K. Eusufzai, and R. E. Mark, "Determination of paper sheet fiber orientation distribution by a laser optical diffraction method," *Journal of Applied Polymer Science*, 1987.

[16] P. P. M. Zuijlen, H. J. C. de Vries, E. N. Lamme, J. E. Coppens, J. van Marle, R. W. Freis, and E. Middelkoop, "Morphometry of dermal collagen orientation by fourier analysis is superior to multi-observer assessment," *Journal of Pathology*, 2002.

[17] J. D. Rubenstein, J. K. Kim, I. Morova-Protzner, P. L. Stanchev, and R. M. Henkelman, "Effects of collagen orientation on mr imaging characteristics of bovine articular cartilage," *Radiology*, 1993.

[18] C. Muehleman, S. Majumdar, A. S. Issever, F. Arfelli, R. H. Menk, L. Rigon, G. Heitner, B. Reime, J. Metge, A. Wagner, K. E. Kuettner, and J. Mollenhauer, "X-ray detection of structural orientation in human articular cartilage," *OsteoArthritis and Cartilage*, 2004.

[19] H. Aghamohammadzadeh, R. H. Newton, and K. Meek, "X-ray scattering used to map the preferred collagen orientation in the human cornea and limbus," *Structure*, 2004.

[20] L. C. Hughes, C. W. Archer, and I. ap Gwynn, "The ultrastructure of mouse articular cartilage: collagen orientation and implications for tissue functionality. a polarised light and scanning electron microscope study and review," *European Cells and Materials*, 2005.

[21] M. J. Clark, "Variation of collagen fiber alignment in a joint surface: A scanning electron microscope study of the tibial plateau in dog, rabbit, and man," *Journal of Orthopaedic Research*, 1991.

[22] S. W. Teng, H. Y. Tan, J. L. Peng, H. H. Lin, K. H. Kim, W. Lo, Y. Sun, W. C. Lin, S. J. Lin, S. H. Jee, P. T. C. So, and C. Y. Dong, "Multiphoton autofluorescence and second-harmonic generation imaging of the ex vivi porcine eye," *IOVS*, 2006.

[23] T. G. Bromage, H. M. Goldman, S. C. McFarlin, J. Warshaw, A. Boyde, and C. M. Riggs, "Circular polarized light standards for investigations of collagen fiber orientation in bone," *The Anatomical Record*, 2003.

[24] I. Singh, *Textbook of Human Histology with Colour Atlas and Practical Guide*. Jaypee, sixth ed., 2010.

[25] B. Cole and A. Gomoll, *Biologic Joint Reconstruction: Alternatives to Arthroplasty*. Slack Incorporated, 2009.

[26] R. A. Stockwell, *Biology of Cartilage Cells*. Cambridge University Press, 1979.

[27] D. J. Taatjes and B. T. Mossman, *Cell Imaging Techniques: Methods and Protocols*. Humana Press, 2006.

[28] G. S. He and S. H. Liu, *Physics of nonlinear optics*. World Scientific, 1999.

[29] C. K. Sun, "Higher harmonic generation microscopy," *Adv Biochem Engin/Biotechnol*, 2005.

[30] J. B. Pawley, *Handbook of Biological Confocal Microscopy*. Springer, third ed., 2006.

[31] P. Török and F. J. Kao, *Optical Imaging and Microscopy: Techniques and Advanced Systems*. Springer, second ed., 2007.

[32] B. Jähne, *Digital image processing*. Springer, sixth ed., 2005.

[33] B. R. Masters and P. T. C. So, *Handbook of Biomedical Nonlinear Optical Microscopy*. Oxford, 2008.

[34] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*. Springer, 1990.

[35] W. Burger and M. Burge, *Digital image processing: an algorithmic introduction using Java*. Springer, 2008.

[36] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, second ed., 2002.

[37] J. Krumm, *Savitzky-Golay Filters for 2D Images*. http://research.microsoft.com/en-us/um/people/jckrumm/SavGol/SavGol.htm, Microsoft Research, 2001, Accessed 2012.

[38] J. O. Smith III, *Mathematics of the Discrete Fourier Transform (DFT)*. BookSurge, 2007.

[39] S. Jayaraman, S. Esakkirajan, and T. Veerakumar, *Digital Image Processing*. Tata McGraw Hill Education Private Limited, 2009.

[40] K. R. Rao, D. N. Kim, and J. J. Hwang, *Fast Fourier Transform - Algorithms and Applications*. Springer, 2010.

[41] J. F. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, 1978.

[42] R. W. Hamming, *Digital Filters*. Dover Publications, third ed., 1998.

[43] V. Imaging, *amira 5, User's Guide*.

[44] S. Lobregt, P. W. Verbeek, and F. C. A. Groen, "Three-dimensional skeletonzation: Principle and algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1980.

[45] C. Pudney, "Distance-ordered homotopic thinning: A skeletonization algorithm for 3d digital images," *Computer Vision and Image Understanding*, 1998.

[46] Y. Zhou and A. W. Toga, "Efficient skeletonization of volumetric objects," *IEEE Transactions on Visualization and Computer Graphics*, 1999.

[47] M. Näf, O. Kübler, R. Kikinis, M. E. Shenton, and G. Székely, "Characterization and recognition of 3d organ shape in medical image analysis using skeletonization," *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*, 1996.

[48] W. C. Ma, F. C. Wu, and M. Ouhyoung, "Skeleton extraction of 3d objects with radial basis functions," *Shape Modeling International*, 2003.

[49] S. S. Abeysinghe, M. Baker, and W. Chiu, "Segmentation-free skeletonization of grayscale volumes for shape understanding," *IEEE International Conference on Shape Modeling and Applications*, 2008.

[50] Z. Yu and C. Bajaj, "A structure tensor approach for 3d image skeletonization: Applications in protein secondary structure analysis," *IEEE International Conference on Image Processing*, 2006.

[51] X. Yuan, J. T. Trachtenberg, S. M. Potter, and B. Roysam, "Mdl constrained 3-d grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images," *Neuroinformatics*, 2009.

[52] Z. Yu and C. Bajaj, "A segmentation-free approach for skeletonization of gray-scale images via anisotropic vector diffusion," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[53] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.

[54] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*.

[55] J. Vince, *Mathematics for Computer Graphics*. Springer, 2010.

[56] Q. Li and J. G. Griffiths, "Least Squares Ellipsoid Specific Fitting," *Geometric Modeling and Processing*, 2004.

[57] J. W. Harris and H. Stocker, *Handbook of Mathematics and Computational Science*. Springer, 1998.

[58] W. H. Steeb, *Problems and Solutions in Theoretical and Mathematical Physics: Introductory level*. World Scientific, second ed., 2003.

Appendix A

Results

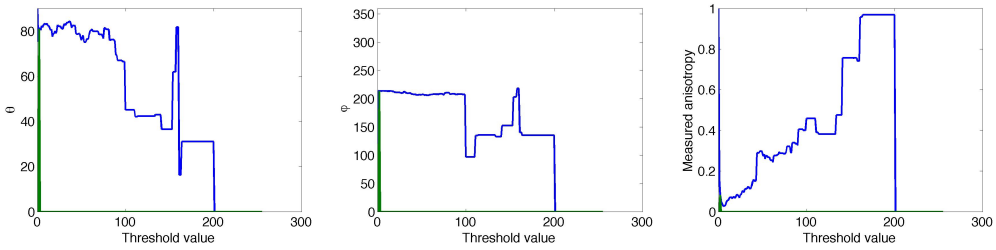
In this section the results are given that are not included in section 4.

A.1 Test 1: Rotational invariance

Table A.1: Rotational test: results ellipsoid method.

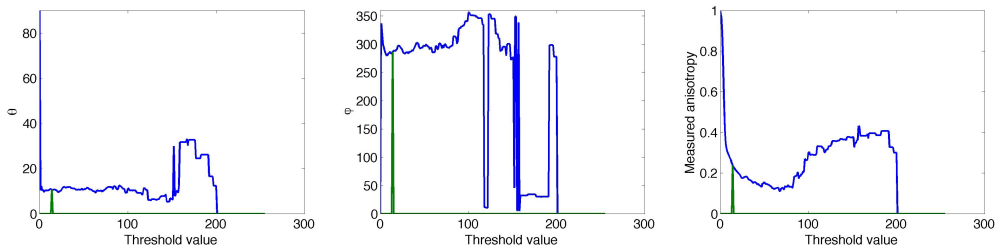
		Data set	1					2		
		Sample	1 (o)	2 (o)	3 (o)	4 (r)	5 (r)	1 (o)	2 (o)	3 (r)
Direction	θ	No rotation	82°	90°	54°	11°	9°	76°	72°	6°
		xy-plane	82°	90°	54°	11°	9°	76°	72°	6°
		xz-plane	82°	90°	54°	11°	9°	76°	72°	6°
	ϕ	No rotation	213°	214°	16°	288°	243°	215°	216°	266°
		xy-plane	213°	214°	16°	288°	243°	215°	216°	266°
		xz-plane	213°	34°	16°	288°	243°	215°	216°	266°
	Anisotropy	No rotation	0.07	0.06	0.83	0.23	0.37	0.20	0.19	0.33
		xy-plane	0.07	0.06	0.83	0.23	0.37	0.20	0.19	0.33
		xz-plane	0.07	0.06	0.83	0.23	0.37	0.20	0.19	0.33

A.2 Test 2: Threshold value dependency



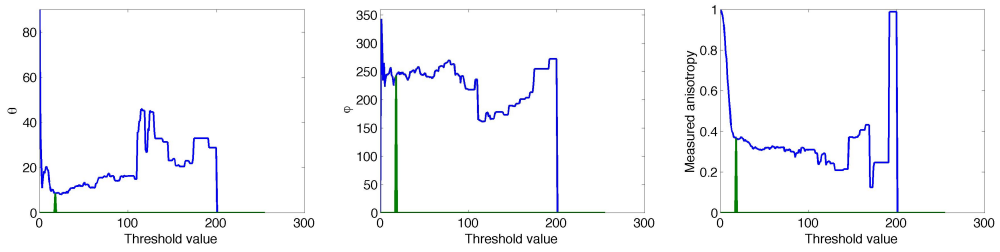
(a) θ (b) ϕ (c) order

Figure A.1: Threshold test: data set 1, sample 1 (o). The green curve shows which threshold value would have been chosen by the program itself.



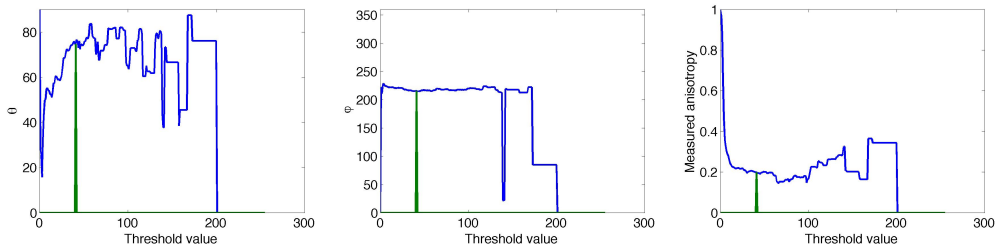
(a) θ (b) ϕ (c) order

Figure A.2: Threshold test: data set 1, sample 4 (r). The green curve shows which threshold value would have been chosen by the program itself.



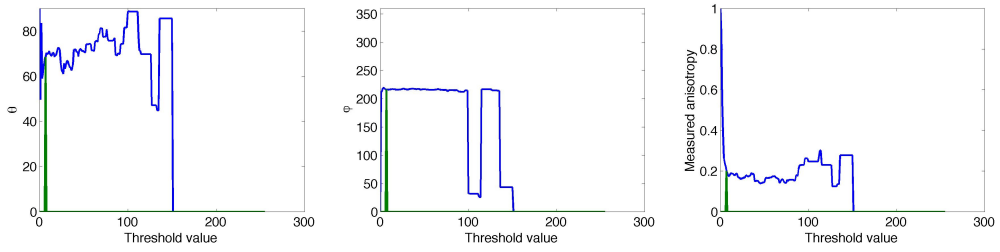
(a) θ (b) ϕ (c) order

Figure A.3: Threshold test: data set 1, sample 5 (r). The green curve shows which threshold value would have been chosen by the program itself.



(a) θ (b) ϕ (c) order

Figure A.4: Threshold test: data set 2, sample 1 (o). The green curve shows which threshold value would have been chosen by the program itself.



(a) θ (b) ϕ (c) order

Figure A.5: Threshold test: data set 2, sample 2 (o). The green curve shows which threshold value would have been chosen by the program itself.

A.3 Test 3: Logarithm versus low frequency filtering

Table A.2: Logarithm vs high frequency filter: results ellipsoid method.

		Data set	1					2		
		Sample	1 (o)	2 (o)	3 (o)	4 (r)	5 (r)	1 (o)	2 (o)	3 (r)
Direction	θ	Logarithm	82°	90°	54°	11°	9°	75°	72°	6°
		LFF*	82°	90°	54°	11°	9°	76°	72°	6°
	ϕ	Logarithm	213°	214°	16°	288°	243°	215°	216°	266°
		LFF	213°	214°	16°	288°	243°	215°	216°	266°
	Anisotropy	Logarithm	0.07	0.05	0.82	0.23	0.36	0.17	0.17	0.32
		LFF	0.07	0.06	0.83	0.23	0.37	0.20	0.19	0.33

* LFF stands for low frequency filtering.

Appendix B

Matlab code

B.1 Point spread function

This code generates a theoretical PSF as described in section 2.2.2.

```
function [PSFmatrix] = PSF(voxelSize, matrixSize)
%The voxel dimensions should be in nm
%The matrix dimensions in number of pixels
% Microscope parameters
lambda=800;     % Wavelength in nm
NA=1.2;         % Numerical aperture
n=1.33;         % Refraction index
p=2;            % p-photon process
%Creating the PSF
syms r;
syms z;
alpha=asin(NA/n);
deltaRho=0.01;
rho=0:deltaRho:1;
v=2*pi*r*n*sin(alpha)/lambda;
u=8*pi*z*n*(sin(alpha/2)^2)/lambda;
J0=besselj(0, (v/p)*rho);
e=exp(1i*(u/p)*rho.*rho/2);
h=sum(e.*J0.*rho)*deltaRho;
PSF=(abs(h))^(p);
maxV=(abs(sum(exp(1i*0*rho.*rho/2).*besselj(0, 0*rho).*rho)*deltaRho))^(p);
PSFnorm=PSF/maxV;
PSFmatrix = zeros(matrixSize(1),matrixSize(2),matrixSize(3));
maxZ=ceil(matrixSize(3)/2);
for m=1:maxZ,
    z=(m-maxZ)*voxelSize(3);
    for k=1:matrixSize(1),
        for l=1:matrixSize(2),
            r=abs(((k-(ceil(matrixSize(1)/2)))*voxelSize(1))...
                +1i*((l-(ceil(matrixSize(2)/2)))*voxelSize(2)));
            t=eval(PSFnorm);
            PSFmatrix(k,l,m)=t;
            PSFmatrix(k,l,(2*maxZ)-m)=t;
        end
    end
end
```

B.2 3-D Savitzky-Golay filter, 4th order polynomial

The Matlab code for the Savitzky-Golay filter applied before skeletonization (step 1 right column in figure (3.1)) is given in this section. The theory of the Savitzky-Golay filter is given in section 2.3.2.

```
function [filteredData] = SGfilter(data,FSx,FSy,FSz)
% FSx, FSy and FSz is the filter size in the x,y and z-direction.
[k,l,m]=size(data);
X=zeros(FSx*FSy*FSz,35);
hFSx=floor(FSx/2);
hFSy=floor(FSy/2);
hFSz=floor(FSz/2);
p=1;
for z=-hFSz:hFSz
    for y=-hFSy:hFSy
        for x=-hFSx:hFSx
            X(p,1)=1;
            X(p,2)=x;
            X(p,3)=y;
            X(p,4)=z;
            X(p,5)=x^2;
            X(p,6)=x*y;
            X(p,7)=x*z;
            X(p,8)=y^2;
            X(p,9)=y*z;
            X(p,10)=z^2;
            X(p,11)=x^3;
            X(p,12)=(x^2)*y;
            X(p,13)=(x^2)*z;
            X(p,14)=x*(y^2);
            X(p,15)=x*y*z;
            X(p,16)=x*(z^2);
            X(p,17)=y^3;
            X(p,18)=(y^2)*z;
            X(p,19)=y*(z^2);
            X(p,20)=z^3;
            X(p,21)=x^4;
            X(p,22)=(x^3)*y;
            X(p,23)=(x^3)*z;
            X(p,24)=(x^2)*(y^2);
            X(p,25)=(x^2)*y*z;
            X(p,26)=(x^2)*(z^2);
            X(p,27)=x*(y^3);
            X(p,28)=x*(y^2)*z;
            X(p,29)=x*y*(z^2);
            X(p,30)=x*(z^3);
            X(p,31)=y^4;
            X(p,32)=(y^3)*z;
            X(p,33)=(y^2)*(z^2);
            X(p,34)=y*(z^3);
            X(p,35)=z^4;
            p=p+1;
        end
    end
end
A=(transpose(X)*X)\ transpose(X);
```

```
A0=zeros(FSx,FSy,FSz);
p=1;
for z=1:FSz
    for y=1:FSy
        for x=1:FSx
            A0(x,y,z)=A(1,p);
            p=p+1;
        end
    end
end
filteredData=zeros(k,l,m);
for x=1:k
    for y=1:l
        for z=1:m
            if (x>hFSx && x<=k-hFSx && y>hFSy && y<=l-hFSy && z>hFSz && z<=m-hFSz)
                for dz=-hFSz:hFSz
                    for dy=-hFSy:hFSy
                        for dx=-hFSx:hFSx
                            filteredData(x,y,z)=filteredData(x,y,z)+data(x+dx,y+dy,z+dz)...
                                *A0(dx+hFSx+1,dy+hFSy+1,dz+hFSz+1);
                        end
                    end
                end
            else
                dxi=0;
                dyi=0;
                dzi=0;
                if (x<=hFSx)
                    dxi=hFSx+1-x;
                end
                if (x>k-hFSx)
                    dxi=(k-hFSx-x);
                end
                if (y<=hFSy)
                    dyi=hFSy+1-y;
                end
                if (y>l-hFSy)
                    dyi=(l-hFSy-y);
                end
                if (z<=hFSz)
                    dzi=hFSz+1-z;
                end
                if (z>m-hFSz)
                    dzi=(m-hFSz-z);
                end
                Xn=zeros(FSx*FSy*FSz,35);
                p=1;
                for zi=-hFSz+dzi:hFSz+dzi
                    for yi=-hFSy+dyi:hFSy+dyi
                        for xi=-hFSx+dxi:hFSx+dxi
                            Xn(p,1)=1;
                            Xn(p,2)=xi;
                            Xn(p,3)=yi;
                            Xn(p,4)=zi;
                            Xn(p,5)=xi^2;
                            Xn(p,6)=xi*yi;
```

```
                    Xn(p,7)=xi*zi;
                    Xn(p,8)=yi^2;
                    Xn(p,9)=yi*zi;
                    Xn(p,10)=zi^2;
                    Xn(p,11)=xi^3;
                    Xn(p,12)=(xi^2)*yi;
                    Xn(p,13)=(xi^2)*zi;
                    Xn(p,14)=xi*(yi^2);
                    Xn(p,15)=xi*yi*zi;
                    Xn(p,16)=xi*(zi^2);
                    Xn(p,17)=yi^3;
                    Xn(p,18)=(yi^2)*zi;
                    Xn(p,19)=yi*(zi^2);
                    Xn(p,20)=zi^3;
                    Xn(p,21)=xi^4;
                    Xn(p,22)=(xi^3)*yi;
                    Xn(p,23)=(xi^3)*zi;
                    Xn(p,24)=(xi^2)*(yi^2);
                    Xn(p,25)=(xi^2)*yi*zi;
                    Xn(p,26)=(xi^2)*(zi^2);
                    Xn(p,27)=xi*(yi^3);
                    Xn(p,28)=xi*(yi^2)*zi;
                    Xn(p,29)=xi*yi*(zi^2);
                    Xn(p,30)=xi*(zi^3);
                    Xn(p,31)=yi^4;
                    Xn(p,32)=(yi^3)*zi;
                    Xn(p,33)=(yi^2)*(zi^2);
                    Xn(p,34)=yi*(zi^3);
                    Xn(p,35)=zi^4;
                    p=p+1;
                end
            end
        end
        An=(transpose(Xn)*Xn)\ transpose(Xn);
        A0n=zeros(FSx,FSy,FSz);
        p=1;
        for zi=1:FSz
            for yi=1:FSy
                for xi=1:FSx
                    A0n(xi,yi,zi)=An(1,p);
                    p=p+1;
                end
            end
        end
        for dz=-hFSz:hFSz
            for dy=-hFSy:hFSy
                for dx=-hFSx:hFSx
                    filteredData(x,y,z)=filteredData(x,y,z)...
                        +data(x+dx+dxi,y+dy+dyi,z+dz+dzi)...
                        *A0n(dx+hFSx+1,dy+hFSy+1,dz+hFSz+1);
                end
            end
        end
    end
end
```

```
end
```

B.3 Fourier transform and frequency filtering

This section contains the Matlab code for step 3 and 4 in figure (3.1). The theory of the Fourier transform and frequency filtering are presented in section 2.4.1 and 2.4.2, respectivly.

B.3.1 3-D Fourier transform and low frequency filtering

```
function [FFTdata] = fft3DImage(data, windowYN, logYN,lffYN)
%windowYN is 1 if a window should be applied before transforming.
%logYN is 1 if the logarithm of the frequency domain should be taken.
%lffYN is 1 if low frequency filtering should be applied.
[k,l,m]=size(data);
if (windowYN==1)
    if (mod(k,2)==1)
        rmax=ceil(k/2);
    else
        rmax=k/2+0.5;
    end
    for x=1:k
        for y=1:l
            for z=1:m
                r=sqrt((x-rmax)^2+(y-rmax)^2+(z-rmax)^2);
                if (r>rmax)
                    r=rmax;
                end
                data(x,y,z)=data(x,y,z)*(cos(r*pi/rmax)+1)/2;
            end
        end
    end
end
FFTdata=abs(fftshift(fftn(data)));
if (logYN==1)
    FFTdata=log(FFTdata+1);
end
if (lffYN==1)
    for x=1:k
        for y=1:l
            for z=1:m
                r=sqrt((x-rmax)^2+(y-rmax)^2+(z-rmax)^2);
                if (r<=2)
                    FFTdata(x,y,z)=0;
                end
            end
        end
    end
end
```

B.3.2 High frequency filtering

```
function [filteredData] = HFF(data, voxelSize, fiberD)
%voxelSize and fiberD are in nm
%fiberD stands for fiber diameter
[k,l,m]=size(data);
fiberDvoxels=fiberD/voxelSize;
radiusImage=ceil(k/fiberDvoxels);
filteredData=subvolume(data,...
   [ceil(k/2)-radiusImage, ceil(k/2)+radiusImage,...
   ceil(l/2)-radiusImage, ceil(l/2)+radiusImage,...
   ceil(m/2)-radiusImage, ceil(m/2)+radiusImage]);
[k,l,m]=size(filteredData);
for x=1:k
    for y=1:l
        for z=1:m
            r=sqrt((x-(radiusImage+1))^2+(y-(radiusImage+1))^2+(z-(radiusImage+1))^2);
            if (r>radiusImage)
               filteredData(x,y,z)=0;
            end
        end
    end
end
```

B.4 Skeletonization

The following codes are the required steps to apply skeletonization as described by section 2.5.2. This is step 2, the right column, in figure (3.1).

B.4.1 Generates a vector field of an image

```
function [VF] = vectorField(data)
[k,l,m]=size(data);
VF=zeros(k,l,m,3);
data=data/max(max(max(data)));
for x=1:k
    for y=1:l
        for z=1:m
            minV=inf;
            for dx=-1:1
                for dy=-1:1
                    for dz=-1:1
                        if (x+dx>0 && x+dx<=k && y+dy>0 && y+dy<=l &&...
                                z+dz>0 && z+dz<=m)
                            if (dx==0 && dy==0 && dz==0)
                            else
                                if (
                                    minV>data(x+dx,y+dy,z+dz))
                                    minV=data(x+dx,y+dy,z+dz);
                                    dxi=dx;
                                    dyi=dy;
                                    dzi=dz;
                                elseif (minV==data(x+dx,y+dy,z+dz))
                                    dxi=dxi+dx;
                                    dyi=dyi+dy;
```

```
                                dzi=dzi+dz;
                            end
                        end
                    end
                end
            end
            if (dxi==0 && dyi==0 && dzi==0)
            else
                VF(x,y,z,1)=(data(x,y,z)-minV)*dxi/...
                    sqrt(dxi^2+dyi^2+dzi^2);
                VF(x,y,z,2)=(data(x,y,z)-minV)*dyi/...
                    sqrt(dxi^2+dyi^2+dzi^2);
                VF(x,y,z,3)=(data(x,y,z)-minV)*dzi/...
                    sqrt(dxi^2+dyi^2+dzi^2);
            end
        end
    end
end
```

B.4.2 Vector field diffusion

```
function [VF] = diffusion(VF, iterations)
for n=1:iterations
    g=weighing(VF);
    VF=vectorDiffusion(VF,g);
end

%WEIGING FUNCTION
function [g] = weighing(VF)
[k,l,m,dim]=size(VF);
g=zeros(k,l,m);
K=1;
for x=1:k
    for y=1:l
        for z=1:m
            c=zeros(3,1);
            for n=1:3
                c(n)=VF(x,y,z,n);
            end
            if (c(1)==0 && c(2)==0 && c(3)==0)
            else
                c=c/sqrt(c(1)^2+c(2)^2+c(3)^2);
                minCos=Inf;
                o=0;
                for dx=-1:2:1
                    if (x+dx>0 && x+dx<=k)
                        s=zeros(3,1);
                        for n=1:3
                            s(n)=VF(x+dx,y,z,n);
                        end
                        if (s(1)==0 && s(2)==0 && s(3)==0)
                            o=1; else
                            s=s/sqrt(s(1)^2+s(2)^2+s(3)^2);
                            d=(c(1)*s(1)+c(2)*s(2)+c(3)*s(3));
                            if (d<minCos)
```

```
                        minCos=d;
                        sn=s;
                    end
                end
            end
            if (y+dx>0 && y+dx<=l)
                s=zeros(3,1);
                for n=1:3
                    s(n)=VF(x,y+dx,z,n);
                end
                if (s(1)==0 && s(2)==0 && s(3)==0)
                    o=1;
                else
                    s=s/sqrt(s(1)^2+s(2)^2+s(3)^2);
                    d=(c(1)*s(1)+c(2)*s(2)+c(3)*s(3));
                    if (d<minCos)
                        minCos=d;
                        sn=s;
                    end
                end
            end
            if (z+dx>0 && z+dx<=m)
                s=zeros(3,1);
                for n=1:3
                    s(n)=VF(x,y,z+dx,n);
                end
                if (s(1)==0 && s(2)==0 && s(3)==0)
                    o=1;
                else
                    s=s/norm(s);
                    d=(c(1)*s(1)+c(2)*s(2)+c(3)*s(3));
                    if (d<minCos)
                        minCos=d;
                        sn=s;
                    end
                end
            end
        end
        if (minCos<Inf && o==0)
            g(x,y,z)=exp(K*(((c(1)*sn(1)+c(2)*sn(2)+c(3)*sn(3))/...
                (sqrt(c(1)^2+c(2)^2+c(3)^2)...
                *sqrt(sn(1)^2+sn(2)^2+sn(3)^2)))-1));
        end
    end
  end
end

%DIFFUSION
function [VFD] = vectorDiffusion(VF,g)
r=1/4;
[k,l,m,dim]=size(VF);
VFD=zeros(k,l,m,dim);
for x=1:k
    for y=1:l
        for z=1:m
```

```
            [xmin,xmax]=limitsDiff(x,k);
            [ymin,ymax]=limitsDiff(y,l);
            [zmin,zmax]=limitsDiff(z,m);
            VFD(x,y,z,:)=VF(x,y,z,:)+r*(g(x,y,z)*[VF(xmin,y,z,:)...
                +VF(xmax,y,z,:)+VF(x,ymin,z,:)+VF(x,ymax,z,:)...
                +VF(x,y,zmin,:)+VF(x,y,zmax,:)-6*VF(x,y,z,:))...
                +(g(xmax,y,z)-g(xmin,y,z))*(VF(xmax,y,z,:)-VF(xmin,y,z,:))...
                /((xmax-xmin)^2)...
                +(g(x,ymax,z)-g(x,ymin,z))*(VF(x,ymax,z,:)-VF(x,ymin,z,:))...
                /((ymax-ymin)^2)...
                +(g(x,y,zmax)-g(x,y,zmin))*(VF(x,y,zmax,:)-VF(x,y,zmin,:))...
                /((zmax-zmin)^2)];
        end
    end
end

%BOUNDARIES function [xmin,xmax] = limitsDiff(x,k)
if (x==1)
    xmin=x;
    xmax=x+1;
elseif (x==k)
    xmin=x-1;
    xmax=x;
else
    xmin=x-1;
    xmax=x+1;
end
```

B.4.3 Generates a skeleton strength map from a vector field

```
function [skeleton] = SSM(VF)
[k,l,m,dim]=size(VF);
skeleton=zeros(k,l,m);
for x=1:k
    for y=1:l
        for z=1:m
            a=0;
            for dx=-1:1
                for dy=-1:1
                    for dz=-1:1
                        if (x+dx>0 && x+dx<=k && y+dy>0 && y+dy<=l &&...
                            z+dz>0 && z+dz<=m)
                            if (dx==0 && dy==0 && dz==0)
                            else
                                a=a+(VF(x+dx,y+dy,z+dz,1)*dx+...
                                    VF(x+dx,y+dy,z+dz,2)*dy+...
                                    VF(x+dx,y+dy,z+dz,3)*dz)/...
                                    sqrt(dx^2+dy^2+dz^2);
                            end
                        end
                    end
                end
            end
            if (a>0)
                skeleton(x,y,z)=a;
            end
```

```
        end
    end
end
```

B.5 Sector method

This section contains the Matlab code required to apply the sector method. The sector method consists of step 5 and 6 in the left column of figure (3.1). The theory behind this method can be found in section 2.3.2 and 2.6.2.

B.5.1 Generates the intensity plot

```
function [I] = sector(data)
dtheta=10/180*pi;
ndtheta=pi/(2*dtheta);
dphi=zeros(ndtheta+1,1);
dphi(1)=2*pi;
for n=1:ndtheta
    dphi(n+1)=2*pi/floor(2*pi/(dtheta/sin((n)*dtheta)));
end
rc=sqrt(2)*dtheta/2;
Ic=zeros(ndtheta+1,2*pi/dphi(ndtheta+1),2);
[k,l,m]=size(data);
r=ceil(k/2);
for x=1:k
    for y=1:l
        for z=1:m
            if (data(x,y,z)>0)
                rn=sqrt((x-r)^2+(y-r)^2+(z-r)^2);
                for n=0:ndtheta
                    for q=1:ceil(2*pi/dphi(n+1))
                        t=n*dtheta;
                        p=q*dphi(n+1);
                        R=[cos(t)*cos(p), cos(t)*sin(p), -sin(t);...
                            -sin(p), cos(p), 0;...
                            sin(t)*cos(p), sin(t)*sin(p), cos(t)];
                        d=[(x-r);(y-r);(z-r)];
                        dnew=R*d;
                        tnew=acos(dnew(3)/rn);
                        if (tnew>pi/2)
                            tnew=pi-tnew;
                        end
                        if (tnew<=rc)
                            Ic(n+1,q,1)=Ic(n+1,q,1)+data(x,y,z);
                            Ic(n+1,q,2)=Ic(n+1,q,2)+1;
                        end
                    end
                end
            end
        end
    end
end
I=zeros(ndtheta+1,2*pi/dphi(ndtheta+1));
for n=0:ndtheta
```

```
    for q=1:ceil(2*pi/dphi(n+1))
        p=q*dphi(n+1);
        for qn=1:ceil(2*pi/dphi(ndtheta+1))
            pn=qn*dphi(ndtheta+1);
            if (pn<=p && I(n+1,qn)==0)
                I(n+1,qn)=Ic(n+1,q,1)/Ic(n+1,q,2);
            end
        end
    end
end
```

B.5.2 Applies the Savitzky-Golay filter to the intensity plot

```
function [phi,theta,der,dif] = SGfilterSector(data,FSx,FSy)
%FSx is the filter size in the x-direction.
%FSy is the filter size in the y-direction.
[k,l]=size(data);
D=zeros(3*k-2,3*l);
hl=floor(l/2);
for x=1:k
    for y=1:l
        if (hl+y<=l)
            ln=hl+y;
        else
            ln=y-hl;
        end
        D(x,y)=data(k-x+1,ln);
        D(x,l+y)=data(k-x+1,ln);
        D(x,2*l+y)=data(k-x+1,ln);
        D(k+x-1,y)=data(x,y);
        D(k+x-1,l+y)=data(x,y);
        D(k+x-1,2*l+y)=data(x,y);
        D(2*k+x-2,y)=data(k-x+1,ln);
        D(2*k+x-2,l+y)=data(k-x+1,ln);
        D(2*k+x-2,2*l+y)=data(k-x+1,ln);
    end
end
D=D/max(max(D));
X=zeros(FSx*FSy,15);
nx=floor(FSx/2);
ny=floor(FSy/2);
p=1;
for y=-ny:ny
    for x=-nx:nx
        X(p,1)=1;
        X(p,2)=x;
        X(p,3)=y;
        X(p,4)=x^2;
        X(p,5)=x*y;
        X(p,6)=y^2;
        X(p,7)=x^3;
        X(p,8)=(x^2)*y;
        X(p,9)=x*(y^2);
        X(p,10)=y^3;
        X(p,11)=x^4;
        X(p,12)=(x^3)*y;
```

```
            X(p,13)=(x^2)*(y^2);
            X(p,14)=x*(y^3);
            X(p,15)=y^4;
            p=p+1;
        end
    end
    A=(transpose(X)*X)\transpose(X);
    F=zeros(FSx,FSy);
    K=zeros(FSx,FSy);
    L=zeros(FSx,FSy);
    p=1;
    for y=1:FSy
        for x=1:FSx
            F(x,y)=A(1,p);
            K(x,y)=A(2,p);
            L(x,y)=A(3,p);
            p=p+1;
        end
    end
    filteredData=zeros(k,l);
    dataDX=zeros(k,l);
    dataDY=zeros(k,l);
    dataD=zeros(k,l);
    for x=1:k
        for y=1:l
            for dy=-ny:ny
                for dx=-nx:nx
                    filteredData(x,y)=filteredData(x,y)...
                        +D(k-1+x+dx,l+y+dy)*F(dx+nx+1,dy+ny+1);
                    dataDX(x,y)=dataDX(x,y)...
                        +D(k-2+x+dx,l+y+dy)*K(dx+nx+1,dy+ny+1);
                    dataDY(x,y)=dataDY(x,y)...
                        +D(k-2+x+dx,l+y+dy)*L(dx+nx+1,dy+ny+1);
                end
            end
            dataD(x,y)=sqrt(dataDX(x,y)^2+dataDY(x,y)^2);
        end
    end
    filteredData=filteredData./max(max(filteredData));
    dtheta=10/180*pi;
    ndtheta=pi/(2*dtheta);
    dphi=zeros(ndtheta+1,1);
    dphi(1)=2*pi;
    for n=1:ndtheta
        dphi(n+1)=2*pi/floor(2*pi/(dtheta/sin((n)*dtheta)));
    end
    [t,p]=find(filteredData==min(min(filteredData)));
    phi=dphi(ndtheta+1)*p*180/pi;
    theta=(t-1)*dtheta*180/pi;
    phi=mean(phi);
    theta=mean(theta);
    if (theta>90)
        theta=180-theta;
        phi=phi-180;
    end
    if (phi<0)
```

```
    phi=360+phi;
end
der=max(max(dataD));
dif=max(max(filteredData))-min(min(filteredData));
```

B.6 Least squares ellipsoid specific fitting

The Matlab code in this section performs the ellipsoid method, which is step 5 and 6 in the right column of figure (3.1). The theory behind this method can be found in section 2.4.2 and 2.6.3.

```
function [phi, theta, anisotropy] = LSESF(data)
[k,l,m]=size(data);
radiusSphere=floor(k/2);
threshold=0;
for x=1:k
    for y=1:l
        for z=1:m
            r=sqrt((x-ceil(k/2))^2+(y-ceil(l/2))^2+(z-ceil(m/2))^2);
            if (radiusSphere<=r && data(x,y,z)>=threshold)
                threshold=data(x,y,z);
            end
        end
    end
end
n=0;
E=zeros(k,l,m);
for x=1:k
    for y=1:l
        for z=1:m
            if (data(x,y,z)>threshold)
                n=n+1;
                E(x,y,z)=1;
            end
        end
    end
end
D=zeros(10,n);
p=1;
for x=1:k
    for y=1:l
        for z=1:m
            if (E(x,y,z)==1)
                xi=(x-ceil(k/2));
                yi=(y-ceil(l/2));
                zi=(z-ceil(m/2));
                D(1,p)=xi*xi;
                D(2,p)=yi*yi;
                D(3,p)=zi*zi;
                D(4,p)=2*yi*zi;
                D(5,p)=2*xi*zi;
                D(6,p)=2*xi*yi;
                D(7,p)=2*xi;
                D(8,p)=2*yi;
                D(9,p)=2*zi;
```

```
                D(10,p)=1;
                p=p+1;
            end
        end
    end
end
q=1048576;
s=0;
while (s==0 && q>=4)
    [v1,v2]=quadricsurface(D,q);
    if (verifyEllipsoid(v1,v2)==1)
        s=1;
    else
        q=q/2;
    end
end
if (s==0)
    q=3.01;
    [v1,v2]=quadricsurface(D,q);
    if (verifyEllipsoid(v1,v2)==1)
        s=1;
    end
end
if (s==1)
    if (q<1048576)
        u=3;
        v=2*q;
        q=(u+v)/2;
        s=0;
        while (((v-u)>=0.1) && s==0)
            if (verifyEllipsoid(v1,v2)==1)
                u=q;
                s=1;
            else
                v=q;
                s=0;
            end
            q=(u+v)/2;
        end
    end
else
    ans='No ellipsoid is found.'
end
A=[v1(1), v1(6), v1(5); v1(6), v1(2), v1(4); v1(5), v1(4), v1(3)];
[eigvectors,eigvalues]=eig(A);
minV=eigvalues(1,1);
for x=2:3
    if (eigvalues(x,x)¡minV)
        minV=eigvalues(x,x);
    end
end
maxV=max(max(eigvalues));
[ROW, COL]=find(maxV==eigvalues);
n=eigvectors(:,COL);
theta=acos(n(3));
phi=atan2(n(2),n(1));
```

```
anisotropy=minV/maxV;

%FINDS v1 AND v2 WHEN THE DATA SET AND q ARE GIVEN
function [v1, v2] = quadricsurface(D, q)
%q is the same as k in section 2.6.3.
S=D*transpose(D);
S11=zeros(6,6);
for x=1:6
    for y=1:6
        S11(x,y)=S(x,y);
    end
end
S12=zeros(6,4);
for x=1:6
    for y=1:4
        S12(x,y)=S(x,y+6);
    end
end
S22=zeros(4,4);
for x=1:4
    for y=1:4
        S22(x,y)=S(x+6,y+6);
    end
end
C1=zeros(6,6);
C1(1,1)=-1;
C1(2,2)=-1;
C1(3,3)=-1;
C1(1,2)=(q/2)-1;
C1(1,3)=(q/2)-1;
C1(2,1)=(q/2)-1;
C1(2,3)=(q/2)-1;
C1(3,1)=(q/2)-1;
C1(3,2)=(q/2)-1;
C1(4,4)=-q;
C1(5,5)=-q;
C1(6,6)=-q;
V=inv(C1)*(S11-(S12*(inv(S22)*transpose(S12))));
[eigvectors, eigvalues]=eig(V);
MAX=max(max(eigvalues));
[ROW, COL]=find(eigvalues==MAX);
v1=eigvectors(:,COL);
v2=-inv(S22)*(transpose(S12)*v1);
v1=v1/abs(v2(4));
v2=v2/abs(v2(4));
if (v1(1)¡0)
  v1=-v1;
  v2=-v2;
end

%VERIFIES THAT THE QUADRIC SURFACE FOUND IS AN ELLIPSOID
function [isEllipsoid] = verifyEllipsoid(v1, v2)
a=[v1(1), v1(6), v1(4), v2(1); v1(6), v1(2), v1(5), v2(2);...
    v1(4), v1(5), v1(3), v2(3); v2(1), v2(2), v2(3), v2(4)];
A=det(a);
b=[v1(1), v1(6), v1(4); v1(6), v1(2), v1(5); v1(4), v1(5), v1(3)];
```

```
B=det(b);
C=v1(1)+v1(2)+v1(3);
D=v1(2)*v1(3)+v1(3)*v1(1)+v1(1)*v1(2)-v1(5)^2-v1(4)^2-v1(6)^2;
pos=0;
if (v1(1)>=0 && v1(2)>=0 && v1(3)>=0 && v2(4)<=0)
   pos=1;
end
if (A<0 && B*C>0 && D>0 && pos==1)
   isEllipsoid=1;
else
   isEllipsoid=0;
end
```

B.7 Data generated ellipsoids and fibres

The Matlab code to make the data generated samples required in section 3.2.4 are given here.

B.7.1 Generates an image containing an ellipsoid

```
function [D] = ellipsoid(imageSize, ellipsoidDim)
%imageSize is just one number, and the generated image is cubic.
%ellipsoidDim contains the length of the three axes.
D=zeros(imageSize,imageSize,imageSize);
r0=ceil(imageSize/2);
theta=-pi/4;
phi=-pi/4;
Ry=[cos(theta),0,-sin(theta);0,1,0;sin(theta),0,cos(theta)];
Rz=[cos(phi),-sin(phi),0;sin(phi),cos(phi),0;0,0,1];
R=Ry*Rz;
for x=1:imageSize
    for y=1:imageSize
        for z=1:imageSize
            Xnew=[1/ellipsoidDim(1);1/ellipsoidDim(2);1/ellipsoidDim(3)]...
                .*(R*[(x-r0);(y-r0);(z-r0)]);
            E=transpose(Xnew)*Xnew;
            r=sqrt((x-r0)^2+(y-r0)^2+(z-r0)^2);
            if (E<=1 && r>2)
               D(x,y,z)=(cos(E*pi)+1)/2;
            end
        end
    end
end
```

B.7.2 Generates an image containing fibres

```
function [F] = fibres(imageSize, radiusF, numberofF, anisotropy)
%radiusF = radius of the fibres
%numberofF = the number of fibres
F=zeros(imageSize,imageSize,imageSize);
for m=1:numberofF
    x0=rand()*(imageSize-1)+1;
    y0=rand()*(imageSize-1)+1;
```

```
    z0=rand()*(imageSize-1)+1;
    theta=anisotropy*rand*pi/2;
    phi=rand*2*pi;
    R=[sin(theta)*cos(phi),sin(theta)*sin(phi),cos(theta)];
    for r0=-imageSize/max(R):imageSize/max(R)
        for z=1:imageSize
            for x=1:imageSize
                for y=1:imageSize
                    if (((x-x0-(r0*R(1)))^2+(y-y0-(r0*R(2)))^2+((z-z0-(r0*R(3)))^2)/4)...
                        <radiusF^2)
                        if (cos(sqrt((x-x0-(r0*R(1)))^2+(y-y0-(r0*R(2)))^2...
                            +((z-z0-(r0*R(3)))^2)/4)*pi/(2*radiusF))>F(x,y,z))
                            F(x,y,z)=cos(sqrt((x-x0-(r0*R(1)))^2+(y-y0-(r0*R(2)))^2...
                                +((z-z0-(r0*R(3)))^2)/4)*pi/(2*radiusF));
                        end
                    end
                end
            end
        end
    end
end
```

B.8 Opening and saving 3-D images

To be able to open the images both with Amira and Matlab they have to be saved as a stack of 2-D tif-files. Therefore a code to both open and save 3-D images as stacks of tif-files had to be made.

```
function [matrixImage] = open3DImage(name, firstSlice, lastSlice)
%The filenames are based on the way they are stored by Amira
if (firstSlice<10)
    firstName=[name '.000' num2str(firstSlice)];
elseif (firstSlice<100)
    firstName=[name '.00' num2str(firstSlice)];
elseif (firstSlice<1000)
    firstName=[name '.0' num2str(firstSlice)];
else
    firstName=[name '.' num2str(firstSlice)];
end
[x y]=size(imread(firstName, 'tif'));
matrixImage=zeros(x,y,lastSlice-firstSlice+1);
z=1;
for (p=firstSlice:lastSlice)
    if (p<10)
        nameSlice=[name '.000' num2str(p)];
    elseif (p<100)
        nameSlice=[name '.00' num2str(p)];
    elseif (p<1000)
        nameSlice=[name '.0' num2str(p)];
    else
        nameSlice=[name '.' num2str(p)];
    end
    matrixImage(:,:,z)=imread(nameSlice, 'tif');
    z=z+1;
end
```

```
function [] = save3DImage(data, saveName, startSlice)
%The files will be saved with numbering like in Amira
data=data/max(max(max(data)));
[k,l,m]=size(data);
for z=1:m
    if (z-1+startSlice¡10)
        name=[saveName '.000' num2str(z-1+startSlice) '.tif'];
    elseif z-1+startSlice¡100
        name=[saveName '.00' num2str(z-1+startSlice) '.tif'];
    elseif z-1+startSlice¡1000
        name=[saveName '.0' num2str(z-1+startSlice) '.tif'];
    else
        name=[saveName '.' num2str(z-1+startSlice) '.tif'];
    end
    imwrite(data(:,:,z), name,'tif');
end
```