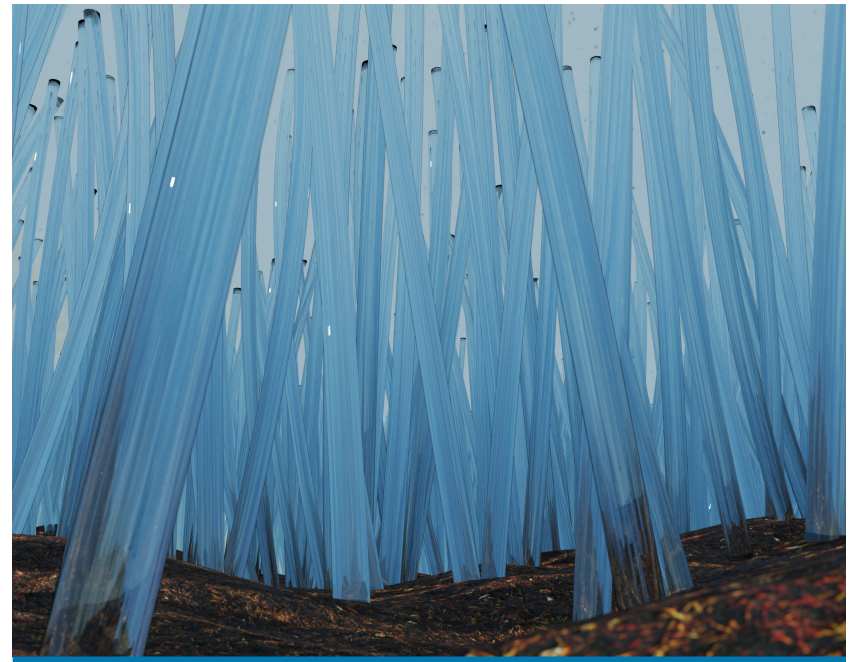
Parametric Fiber Analysis for Glass Fiber-reinforced Composite Tomographic Images

Tim Elberfeld

Parametric Fiber Analysis for Glass Fiber-reinforced Composite Tomographic Images

Tim Elberfeld



Supervisors **Prof. Dr. Jan Sijbers | Prof. Dr. Ir. Jan De Beenhouwer**

Thesis submitted for the degree of Doctor of Science: Physics
Faculty of Science | Antwerp, 2023

University of Antwerp

UNIVERSITY OF ANTWERP

DOCTORAL THESIS

Parametric Fiber Analysis for Glass Fiber-reinforced Composite Tomographic Images

Author:
Tim ELBERFELD

Supervisors:
Prof. Dr. Jan SIJBERS
Prof. Dr. Ir. Jan DE BEENHOUWER

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Sciences

in the

Vision Lab
Deptartment of Physics

June 6th, 2023

Doctoral committee:

Prof. Dr. Joke Haderman
Prof. Dr. Joris Dirckx
Prof. Dr. Jan Sijbers
Prof. Dr. Ir. Jan De Beenhouwer

External Jury:

Prof. Dr. Katja Schladitz
Prof. Dr. Christoph Heinzl

Cover art: "Glass Forest"
Render of a population of curved fibers generated using the simulation algorithms described in this thesis. Modeled and rendered using Blender [1] and the Cycles rendering engine.

Contact information:

✉ **Tim Elberfeld**
Vision Lab, Dept. of Physics
University of Antwerp (CDE)
Universiteitsplein 1, Building N1.14
B-2610 Wilrijk, Antwerpen, Belgium
✉ tim.elberfeld@web.de
http://visielab.ua.ac.be/people/tim-elberfeld
https://orcid.org/0000-0002-0701-2421

Declaration of Authorship

I, Tim ELBERFELD, declare that this thesis titled, "Parametric Fiber Analysis for Glass Fiber-reinforced Composite Tomographic Images" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Have no fear of perfection - you'll never reach it."

Salvador Dalí

"A society grows great when old men plant trees whose shade they know they shall never sit in."

Greek proverb

Acknowledgements

When I was looking for a PhD position back in 2017 I was fresh out of university. I had just finished my master degree, for which I wrote my thesis at the European Synchrotron Radiation Facility. Sparked by seeing all the people during their own PhD programs I decided to go for one as well. Initially I looked for places in Germany, trying to find something somewhat close to home. But it soon became clear that there were more attractive opportunities elsewhere. Without thinking much about it I wrote an email to prof. Sijbers, asking to get a position on the "Metroflex" project, which dealt with metrology in projection space. It didn't even take four hours before I received an answer asking me for a video call (*"Are you free in about one hour?"*). A week later I had been accepted for a different project, which dealt with the analysis of glass fiber-reinforced polymers in collaboration with an Austrian university. This was an exciting time for me and I quickly arranged everything to move to Belgium. Little did I know then the implications this move would have. While all these exciting new impressions rained down on me and I was finding my way in a new country with a new language.

As I am writing this my life is very different than it was back then. Not only did the PhD pose its challenges at times, there was also a period of a global pandemic, that changed life in significant ways and with incredible speed. Before that I had the opportunity to travel to America and Australia for the first time and met scientists from all over the world and see research facilities everywhere I went. I learned how to communicate my findings at conferences and how to work on a project with a timeline of several years without despairing before the sheer amount of work ahead.

I am currently sitting in an apartment that will be my home for another couple months before I move to a house that my partner Marie and I bought. Belgium has become my home, though Germany will always be as well. I now speak Dutch every day at work. I am proud that I managed to take on the challenge of this behemoth of a task that is the PhD and not get chewed up and spit out. It was a time of a lot of development, both professionally and personally. It formed my way of thinking and consolidated my ideals about what I deem important.

As every single person that takes on a project of this magnitude, I did of course not do it on my own. Working in the Vision Lab is a constant collaboration and exchange of ideas. Thank you to my supervisors Jan Sijbers and Jan De Beenhouwer for accepting my application and to take a chance

on me. I am grateful for the opportunities you gave me and the help in navigating the PhD, even when the waters were not always still. Thank you also to the project partners at Univesity of Applied Sciences Upper Austria, in particular Christoph Heinzl and Bernhard Fröhler it was great to be able to contribute to common research and see the work take shape. Thank you also to my friends and direct colleagues at Vision Lab: Jonathan, Nathanaël, Árpád, Daniel, Van, Alice, Jens and many more. Thank you for the elaborate discussions about important and not so important topics and evenings spent in bars or playing games. And of course a very big thank you to my family and my Marie. Without you I could have never finished this journey. Thank you for the many laughs and cries, the moral support and always having my back. Thank you to my mother and Klaus for providing me a home away from home and that I can always count on you. Thank you Julia for always being a supportive sister, even if the physical distance between Antwerp and Munich prevents us from seeing each other as often as I'd like. Thank you to Vinciane and Matthieu and Nicolas, Astrid, Lucie and Matteo for being my second family. And lastly and importantly: thank you Marie. You are a great partner in crime and without your support I am sure I would have given up on this project a long time ago. Thank you for having my back, thank you for being there for me for every step of the way.

I am lucky to have you all in my life!

T. Elberfeld

Tim Elberfeld
Antwerp, January 2023

Contents

Summary

Fiber-reinforced composites are an important part of modern material science. They are used in every area of our lives in some shape or form. Car manufacturers use them to make the car body lighter and give it better compression properties if there is an accident. Concrete mixed with fibers is now a widely used material not only in construction, but also in private home renovation and improvement. Wind turbines have blades made of plastic, reinforced with longer strands of glass fibers to make them lighter and stronger against the forces of the wind. The more the field progresses, the more the composites can be improved and their applications diversified. A fiber-reinforced composite consists of a base or *matrix* material that embeds some sort of fibrous material, which is the reinforcement component. This can be glass or carbon fibers, steel bars or natural fibers like bamboo. They can be added in layers, bundles, weaves or loosely dispersed within the material before its set.

A preferred method for investigating composites is X-ray computed tomography. This technique can generate a fully 3-dimensional image, that reveals the inside structure of the analyzed samples. As X-rays penetrate most materials easily, it is possible to acquire images of the attenuation of the incoming radiation when placing such a material sample between the source and the imaging device. Several of those so-called projections are collected over a large angular range, preferably 180° or more. Using mathematical methods the volumetric image, the reconstruction, can then be computed from the projections.

The thesis you are reading is dealing with the problem of the quantitative and statistical analysis of such composites, using X-ray computed tomography. More precisely, it shows approaches to detect the fibers in volumetric images of glass fiber-reinforced polymers and analyze their structure to give insight into their geometrical properties. After introducing the foundations for the work, a first framework for the extraction and analysis of straight fibers, called PARE, is introduced. This framework is then extended to also deal with fibers with arbitrary curvature. Parametric models for both straight and curved fibers are presented and it is shown how those parametric models can be improved using numerical optimization based on the information in the X-ray projection images.

Samenvatting

Vezelversterkte composieten vormen een belangrijk onderdeel van de moderne materiaalwetenschappen. Ze worden op elk gebied van ons leven in een of andere vorm gebruikt. Autofabrikanten gebruiken ze om de carrosserie lichter te maken en betere compressie-eigenschappen te geven bij een ongeval. Beton gemengd met vezels wordt niet enkel in de bouw veel gebruikt, maar ook bij renovaties en verbeteringen van particuliere woningen. Windturbines hebben wieken uit kunststof, versterkt met langere strengen glasvezel om ze lichter en sterker te maken tegen de krachten van de wind. Hoe meer het veld vordert, hoe meer de composieten kunnen worden verbeterd en hun toepassingen kunnen worden gediversifieerd. Een vezelversterkte composiet bestaat uit een basis- of *matrix*-materiaal waarin een soort vezelachtig materiaal is ingebed, wat de versterkingscomponent is. Dit kunnen glas- of koolstofvezels, stalen staven of natuurlijke vezels zijn zoals bamboe. Ze kunnen in lagen, bundles, of weefsels worden toegevoegd of los van elkaar in het materiaal worden vespreid voordat het hard wordt.

Een voorkeursmethode voor het onderzoeken van composieten is röntgencomputertomografie. Deze methode kan een volledig driedimensionaal beeld genereeren, dat de interne structuur van geanalyseerde monsters onthult. Aangezien röntgenstralen de meeste materialen gemakkelijk doordringen, is het mogelijk om beelden te verkrijgen van de verzwakking van de binnenkomende straling wanneer een dergelijk materiaalmonster tussen de bron en het beeldvormingsapparaat wordt geplaatst. Verschillende van die zogenaamde projecties worden verzameld over een groot hoekbereik, bij voorkeur 180° of meer. Met behulp van wiskundige methoden kan vervolgens het volumetrische beeld, de reconstructie, worden berekend uit de projecties.

Het voorgelegde proefschrift behandelt het probleem van de kwantitatieve en statistische analyse van dergelijke composieten met behulp van röntgencomputertomografie. Meer precies toont het benaderingen om de vezels te detecteren in volumetrische beelden van glasvezelversterkte polymeren en hun structuur te analyseren om inzicht te krijgen in hun geometrische eigenschappen. Na de introductie van de basis voor het werk, wordt een eerste framework voor de extractie en analyse van rechte vezels, PARE genaamd, geïntroduceerd. Dit framework wordt vervolgens uitgebreid om ook vezels met een kromming aan te pakken. Parametrische modellen voor zowel rechte als gebogen vezels worden gepresenteerd en er wordt getoond hoe die parametrische modellen kunnen worden verbeterd met behulp van numerieke optimalisatie op basis van de informatie in de röntgenprojectiebeelden.

Chapter 1

Introduction

The development of advanced composite materials consisting of a bulk material component and one or more reinforcing components is an essential part of material science in its quest of making existing materials stronger and lighter [2]. During the development process, researchers and engineers need to know what properties a newly created material has or will have. Traditionally, new materials are analyzed in a process called non-destructive testing (NDT). This can be using load testing, for example, where a small sample of the material is created and then twisted, pulled, pushed or otherwise deformed and the forces and mechanical stresses that are developed in the testing machines are recorded [3]. This process seamlessly transitions into destructive testing, where the material sample can actually be destroyed during these tests. Some methods also grind small sections off of a material and pictures are taken of the revealed surface to get more information about the structure inside the sample [4]. However, this destructive way of retrieving the information can deform the material sample in a way that may yield wrong conclusions.

The preferred methods of getting information about a material in a in a non-destructive way are visual in nature. The topic of this thesis is one of these visual methods, X-ray micro-computed tomography (μCT). It enables the visualization of the three dimensional structure of a material on the micron level [5]. During a μCT scan, an X-ray beam is directed at a sample of a material to investigate. Acquiring images of these X-rays from several known directions enables the reconstruction of a 3D volumetric image that contains information about the internal structure of the sample. μCT can reveal the fine structures that influence the properties of the material significantly, while being non invasive and flexible to accommodate various experimental setups.

Traditionally, the material properties are investigated by applying a pipeline of image processing operations to the resulting tomographic images of the materials. The behavior of materials under force loads typical for the desired application, can be also be investigated using so-called representative elementary volumes or representative volume element (RVE). A RVE is the smallest (simulated) volume that can be used for a virtual force measurement experiment that will result in the same measurement as the measurement on the real material. The representative elementary volumes can be created based on the statistical information retrieved via μCT.

As the demand for lighter, more complex (multi-)material objects grows, better tools are required to investigate their properties. This includes the need for increasingly higher resolution scans with pixel sizes in the low micrometer range, which produces vast amounts of data. While in the past the visual interpretation of images was commonplace, material science has now gravitated towards quantitative imaging. One reason for that is to reduce the amount of data while preserving the accuracy achieved by the high resolution of modern imaging setups: A few parameters, estimated from a high resolution 3D volume, can be stored more easily than the amount of voxels that represent the same object with comparable resolution. Especially objects of interest with a large aspect ratio, such as glass fibers embedded in a resin matrix, can benefit from moving from voxel based representations to model based ones. Models can take various forms, from single scalar parameters modeling a distribution of values, to entire mesh models of objects contained in a volume. The central question for the research presented in this thesis was:

How can a feedback loop be inserted into the processing pipeline used to generate these models?

The typical image processing workflow is unidirectional and comprised of the components acquisition, reconstruction, processing, and analysis. these techniques can be used to derive data such as the orientation distribution of fibers, the size distribution of included air bubbles and many other values. However, this workflow allows erroneous estimations to cascade into further errors later on. Using a feedback mechanism the estimates can be the basis to initialize a guess for the model, which then serves as the *a priori* knowledge for another round of estimation.

The work presented here shows one possible answer to this question in the context glass fiber-reinforced polymer (GFRP). Different models are presented and their parameters estimated using image processing. A feedback mechanism, which iteratively refines the model estimates, is then introduced. The thesis begins with a more detailed explanation of the methods used within the presented work. The following chapters introduce said work, from modeling straight fiber composites towards more flexible models of curved fibers and their application to both simulated data and realistic datasets.

Chapter 2

Foundation

Computed tomography (CT) is an essential technique in NDT. It enables the imaging of the microstructure of materials and thus their computational analysis. This chapter introduces the foundations for this imaging technique by explaining the production of the X-rays, followed by an explanation about how this radiation interacts with matter (see Section 2.2). During the imaging process, this interaction with matter is recorded by detecting the X-rays passing through an object with a light sensitive detector. The resulting images, so-called projections or radiographs, contain information about the inner structure of the object (see Section 2.3). This structure can be reconstructed using the X-ray transform [6] which has a close relation to the Radon transform. Several reconstruction algorithms based on the X-ray transform have been proposed and are explained in Section 2.4. Reconstruction leads to a visualization of the material structure, like the sample of rockwool shown in Figure 2.1. Section 2.5 deals with the manufacturing of advanced materials and how the structures of those materials can be analyzed using image processing techniques and NDT methods, focusing on X-ray CT as the main method of NDT. The contents of sections 2.2 and 2.3, including the figures, are largely based on the author's unpublished master's thesis [7].

2.1 Non-destructive Testing

Early microscopists interested in the inner structure of plants, seeds, raw building materials, snow, small animals and even humans started preparing their subjects in a way that enabled the cutting of thin slices or creation of flat faces. At first these slices were the only thing to be investigated and techniques were developed to gain quantitative information from these sections. For example, steel used to be (and still is) characterized by grinding and polishing a flat face and then taking images of that face to analyze the content

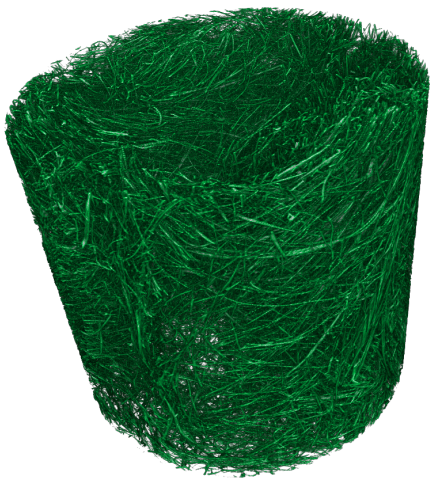


FIGURE 2.1: Example of a volume render of a X-ray computed tomography (XCT) reconstruction of a fibrous material - rockwool. The scan was taken using the FlexCT scanner at the VisionLab [8]. The sample was kindly provided by Jonathan Sanctorum from his roof insulation.

of carbon in the steel [4]. The resulting images of material slices, however, could also be stacked together to form a fully volumetric image of the sample with the downside of having to not only destroy it in the process, but the preservation process also potentially modifying the sample, sometimes significantly [9, 10]. This was the foundation of the field of stereology [11], which aims to gain quantitative measurements of material structures. Today, non-destructive testing of materials via various imaging techniques or other measurements replaces the destructive slice by slice analysis [12]. Some of the techniques used to that end are presented subsequently.

The visual inspection of parts by means of visible light camera images is by now the industry standard for quality control. Applications include the investigation of welds [13], inspection of printed circuit boards (PCBs) [14] and the testing of the integrity of the packaging of medications [15].

Fluorescence spectroscopy is a technique used in materials science, biochemistry and biophysics to investigate the chemical composition of materials by their emitted light spectra [16]. With this technique it is possible to sequence DNA [17] or localize functional molecules in cells [16].

Different electromagnetic spectra can also be exploited by extending the

visual range of digital cameras in hyperspectral imaging techniques. Here, the camera is sensitive to infrared and/or ultraviolet light and can give additional information about the observed specimen. This can be used, among other things, to visually test the ripeness of fruit [18] or monitor the growth of plants under different conditions [19].

Microscopy can of course be used in non-destructive ways as well. Confocal (X-ray) laser microscopy provides 2- or 3-dimensional visualization of material surfaces and internal structure [20]. Using electron microscopy atoms in materials can be localized and the structure of nano-particles characterized [21].

For the investigation of mechanical properties of materials, one can apply forces in what is called load testing [22] or a representative mathematical model can be created that enables the simulation of the resulting mechanical properties via finite-element analysis [23]. The model itself can also be used to gain statistical information about material distribution and other relevant features. Often such a representative model can be estimated from scans of a material sample using X-ray computed tomography (XCT), which is the subject of this thesis.

2.2 X-rays

X-radiation is a form of electromagnetic radiation that has a wavelength in the order of roughly between 10^{-8} m and 10^{-13} m . In the beginning stages of the field of research, they were generated using an X-ray tube (Figure 2.2). This is a vacuum glass tube containing a cathode and an anode (or anti-cathode).

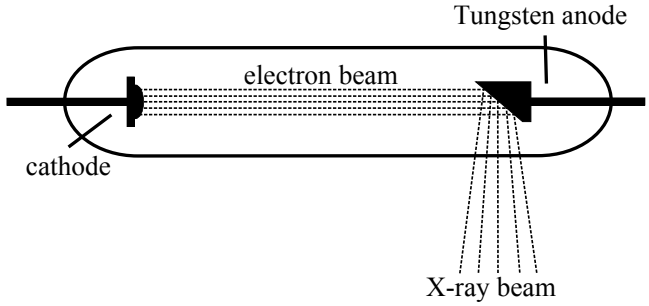


FIGURE 2.2: Schematic of an X-ray tube. The electrons hit the anode and are decelerated by the material. This generates X-ray radiation that is emitted from the tube.

A voltage of 20 to 600 kV accelerates the electrons from a heated cathode towards an anode [24]. Hitting the anode the electrons decelerate rapidly during entry into the first few micrometers of the anode surface. This rapid deceleration causes the emission electromagnetic radiation in form of X-rays.

The photons generated this way are made up of two types of radiation, called Bremsstrahlung and characteristic radiation. Together those form what is generally known as X-rays [24, 25, 26].

2.2.1 Bremsstrahlung

The Bremsstrahlung, i.e. the continuous part of the spectrum, as seen in Figure 2.3, can be calculated by Kramers' law, which gives a function $I_0(\lambda)d\lambda$ expressing the intensity I_0 of the spectrum in a wavelength interval $d\lambda$:

$$I_0(\lambda)d\lambda = K\left(\frac{\lambda}{\lambda_{\text{min}}} - 1\right)\frac{1}{\lambda^2}d\lambda, \tag{2.1}$$

where λ is the average wavelength in the interval, λ_{min} is the minimum wavelength computed from (2.2) and the constant K is an empiric constant, proportional to the atomic number Z of the anode element [27]. Its name derives from the German "bremsen" (English: to brake), because it is emitted as a result of the deceleration.

The minimum wavelength of that spectrum is given by the Duane-Hunt law

$$\lambda_{\text{min}} = \frac{hc}{e\,U_{\text{cathode}}}, \tag{2.2}$$

where e is the charge of an electron. U_{cathode} is the voltage of the cathode used to produce the electron beam [28]. The constants c and h are the speed of light in vacuum and Planck's constant, respectively. This means that the only parameter that can be changed in order to influence the minimum wavelength, is the cathode voltage. The only thing that is influenced by the material, is the spectral intensity, but not the wavelength distribution, as seen in Eq. (2.1), where K is only a scaling factor.

The other part of the X-ray spectrum is composed of individual peaks at different wavelengths, which are specific to the anode material. Since these peaks can be used as a signature of an element, this radiation is called characteristic radiation. There are two characteristic peaks, that can be observed in an X-ray spectrum.

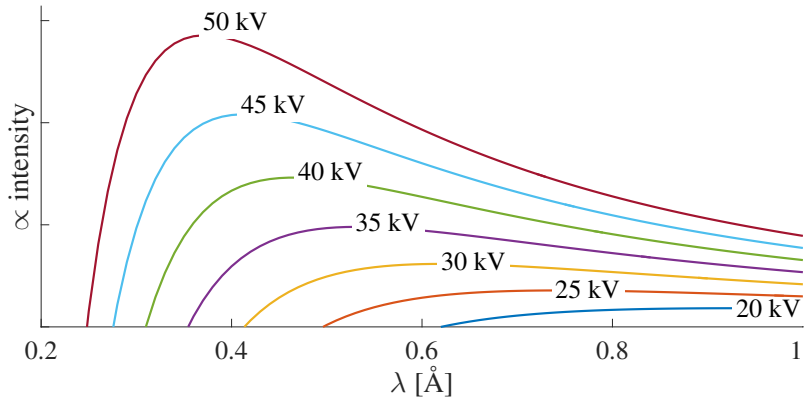


FIGURE 2.3: Resulting theoretical Bremsstrahlung spectra using different anode voltages, with the atomic number $Z = 74$ (Tungsten). The y-axis shows arbitrary units proportional to the intensity.

2.2.2 Characteristic Radiation

The peaks of the characteristic radiation spectrum correspond to the K_α and K_β peaks of the respective element. The K_α peak corresponds to radiation emitted when an electron releases energy while transitioning from the L to the K shell. The K_β similarly corresponds to the energy released in the transition from the M to the K shell[25].

The frequency f_α at which these peaks occur for the K_α line for example can be calculated by Moseley's law

$$f_\alpha = \frac{3}{4} R \left(Z - 1\right)^2 \tag{2.3}$$

with Rydberg's constant expressed as a frequency $R = 3.29 \cdot 10^{15}$ Hz [29].

2.2.3 Refractive Index for X-rays

In traditional optics we learn that the refractive index of the least dense medium (a vacuum) is $n = 1$ and for every other medium that refracts light it is $n \geq 1$. Moreover, it is common knowledge that there are only a few materials that are visually transparent, that is they let radiation in the visible light spectrum pass through. For X-rays however, the refractive index is complex valued

and defined as

$$n^* = 1 - \delta + \mathrm{i}\beta, \tag{2.4}$$

where $1 - \delta$ is the real part, with

$$\delta = \frac{1}{2\pi} r_e \rho_e \lambda^2, \tag{2.5}$$

and β

$$\beta = \frac{\mu_{\mathrm{ph}} \lambda}{4\pi}, \tag{2.6}$$

denotes imaginary part, representing an alternative notation of the usual refractive index n. The real part of the refractive index is usually smaller than 1. The decrement δ is composed of the wavelength λ, the electron density ρ_e of the material and the radius of the electron $r_e = 2.818 \cdot 10^{-15}$ m. The parameter μ_{ph} in (2.6) is the photoelectric absorption coefficient [25]. Due to its high energy most materials behave like glass does for visible light from the perspective of X-rays, so refraction occurs in most materials acccording to Snell's law. The behavior of the refraction however, is altered compared to visible light, because of the refractive index being lower than 1. An example for that would be the direction of a beam of light entering a denser medium. In visible light optics the expected outcome would be that the beam of light will be refracted towards the perpendicular/surface normal. For X-ray optics, the reverse is true – i.e. the beam will be refracted *away* from it.

2.3 Tomographic Projection

For now the properties of X-rays have only been discussed in general terms of their production and how they function within the context of optics. For tomographic applications it is however important to also understand the interaction of the specific range of electromagnetic radiation with matter more closely.

2.3.1 Lambert-Beer law

The Lambert-Beer law describes by how much the radiation penetrating a medium gets attenuated while traveling a distance l through this medium

$$I(l) = I_0 \mathrm{e}^{-\int \mu(l) \mathrm{d}l}. \tag{2.7}$$

The attenuation of the material is given by its material dependent attenuation coefficient μ, which varies depending on the location in the object [6]. We can see that given no object and therefore no attenuation, the equation yields I_0, which is the intensity of the X-ray beam. Of course from (2.1), it can be seen that this intensity largely depends on the energy used to generate the X-rays. In this thesis the above simplification of the Lambert-Beer law is sufficient, but there exist works that deal with a full spectrum dependent initial intensity. That model is commonly referred to as the polychromatic forward model [30]. Consequently, Equation 2.7 is called the monochromatic forward model. A direct result of the polychromatic nature of the incident radiation is a skewing of the spectrum that arrives at the detector compared to the radiation exiting the source. This process is called "beam hardening", because it skews the energy distribution towards the higher energy (also referred to as "harder") X-rays, because the lower energy part of the spectrum is either more absorbed by the material or refracted away from the detector.

2.3.2 Measuring X-ray Intensities

Modern (visible) light detectors, as they can be found in cameras and phones, are essentially arrays of silicone photodiodes [31]. Due to the better signal-to-noise ratio (SNR), silicone-based detectors are most often used and are the furthest evolved technologically. However, the response of the silicone diodes is in the wavelength range of 190-1100 nm [32], which is several orders of magnitudes higher than the wavelength of X-rays around 0.25 nm. An easier solution than making a photodiode that is sensitive in the X-ray spectrum is to use an existing detector sensitive in the visible spectrum and converting the X-rays to visible light [6]. One way to convert X-rays into visible light is to use their ionizing property and ionize a gas, which then becomes luminescent. This principle is also used in the Geiger-Müller counter [33]. This method is not used in many detectors anymore due to the low conversion efficiency. The more useful method is the use of a solid scintillator material [6]. These materials can be manufactured as thin slabs and then attached to a photon detector. The X-rays entering the medium, often made from cesium iodide, bismuth germanate or cadmium tungstate, generate luminescent light. The material absorbs the incoming radiation and emits radiation of a longer wavelength in a process called scintillation, which gives the materials their name. The choice of material influences the time of the "after glow", sensitivity to incoming wavelength, spectrum of the emitted light and quantum efficiency.

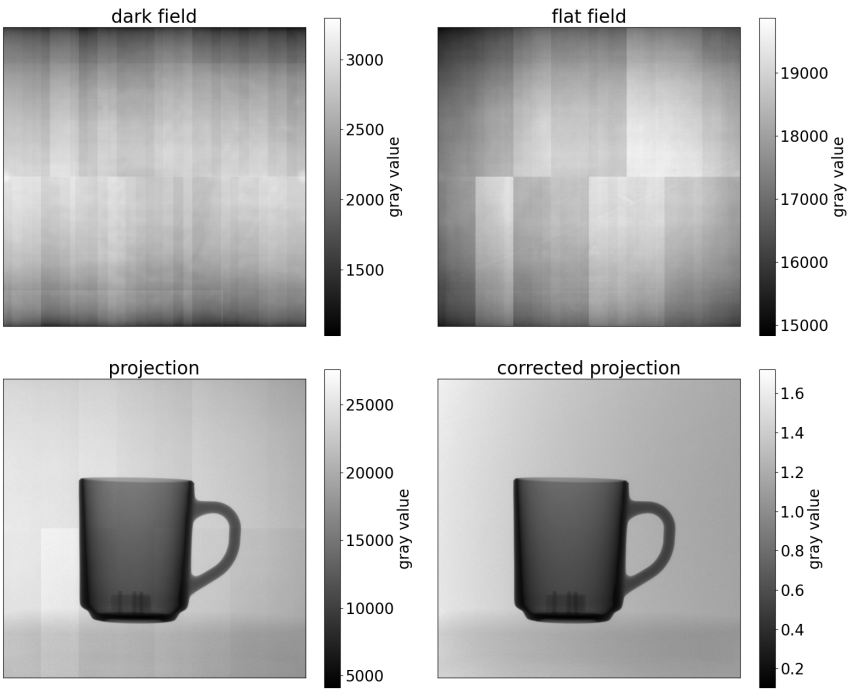


FIGURE 2.4: Acquisition of a detector image without illumination (darkfield, top left), without an object but with illumination (flatfield, top right), X-ray projection (bottom left) and illumination/flatfield corrected projection (bottom right) of a coffee mug, scanned during one of the experiments at the FlexCT scanner [8].

2.3.3 Flatfield and Darkfield

Knowing how one ray behaves and interacts with an object is great, but to obtain insights into an object's density differences, the attenuation coefficient needs to be related to how the intensity of the attenuated light changes. Additionally the intensity has to be sampled in various places. This requires

solving Equation 2.7 for $\int \mu(l)\mathrm{d}l$ for all detector pixels in a recorded projection image.

$$\int \mu(l)\mathrm{d}l = -\ln\left(\frac{I_p(l)}{I_{0,p}}\right), \forall p \in \boldsymbol{p} \tag{2.8}$$

$$\boldsymbol{p} = -\ln\left(\frac{I}{I_0}\right). \tag{2.9}$$

The incident intensity $\boldsymbol{I}_0 : \mathbb{R}^2 \mapsto \mathbb{R}$, the beam intensity when unobstructed by any object, has to be measured before, during, or after the experiment. The above equation is often referred to as "log correction", because of using the natural logarithm. The incident intensity is called the "flatfield", so the term "flatfield correction" is used interchangeably. Due to the nature of the digital detectors used to record the X-ray beam, this does not fully correct the measured image, though. Detectors exhibit noise, even when no light is being recorded. The pixels can also have an erroneous constant offset that is always detected by the detector, shifting the incident intensity by that amount. This thermal noise and the constant offsets can be recorded and also used to correct the image. Because it is measured with the X-ray beam turned off, this is referred to as the *darkfield* $\boldsymbol{I}_d$. Thus, Equation 2.8 becomes

$$\boldsymbol{p} = -\ln\left(\frac{\boldsymbol{I} - \boldsymbol{I}_d}{\boldsymbol{I}_0 - \boldsymbol{I}_d}\right). \tag{2.10}$$

Additionally some detector pixels might be defective, signaling to the computer that the pixel did not record any intensity, or that its intensity was at the maximum level. These pixels are referred to as dead pixels and hot pixels, respectively. To correct for these errors, the median value in a region around each of those pixels can be inserted to minimize the wrong pixel's influence. This process is visualized in Figure 2.4.

With (2.10) the spatial density distribution of the object can be reconstructed from several projection images acquired from different angles. This can be either done with a line detector that rotates around the object, or with a 2D detector, as described in Subsection 2.3.2. The former will result in a reconstruction of a slice through the object of interest, the latter in a full three dimensional reconstruction.

2.3.4 Noise

Due to the type of detectors used, the acquired images will exhibit noise. Noise is part of any image acquired with an imperfect means of recording incident intensity. It appears in images as local variations of recorded intensity with a magnitude related to the SNR. These variations are a result of the discretization of the incident radiation during the acquisition process, which causes different amount of photons hitting adjacent detector pixels even when exposed to similar flux of radiation.

An accurate statistical model of the noise is called Poisson noise (also known as shot noise) [34]. The Poisson distribution

$$\mathrm{Pf}(N = k) = \frac{\lambda_e^k e^{-\lambda_e}}{k!}, \tag{2.11}$$

with k discrete events and an expected value of λ_e can model the process of k photons with an expected wavelength of λ_e, which is more in line with the physical process than the Gaussian model often used when modeling noise. A Poisson distribution with a sufficiently high amount of discrete events approximates a Gaussian distribution with a mean equal to the expected value λ_e of the Poisson distribution.

A key difference with shot noise versus Gaussian noise is that both the event count and the expected value change based on the incident intensity. That is, an object in the beam not only attenuates the X-rays, reducing the amount of photons hitting the specific spot of the detector ”behind’ it, it also skews the energy distribution due to beam hardening, as explained in Section 2.3.1. In many cases it can still be sufficient to model the noise using a Gaussian distribution.

2.3.5 X-ray Transform

For the three dimensional volumetric reconstructions of samples a method called tomography is employed. It uses the properties of a mathematical transform called the Radon transform [35] which can be expressed mathematically in the following way:

$$\mathfrak{R}f(\boldsymbol{s}, \vartheta) = \int_{-\infty}^{\infty} f(\boldsymbol{s}\sin\vartheta + \boldsymbol{t}\cos\vartheta, -\boldsymbol{s}\cos\vartheta + \boldsymbol{t}\sin\vartheta)d\boldsymbol{t}, \tag{2.12}$$

where $\boldsymbol{s} \in \mathbb{R}^n$ and $\vartheta \in [0, \pi)$.

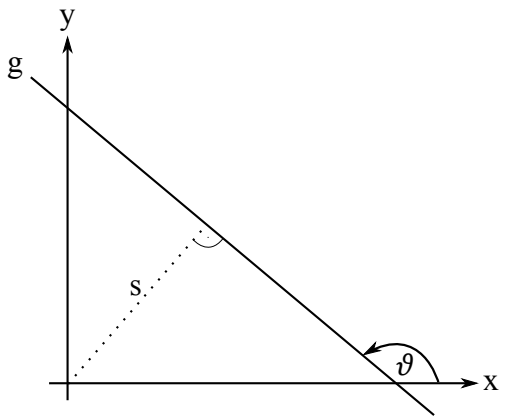


FIGURE 2.5: Line parameters as used in the (2D) Radon transform.

Equation (2.12) is a special case of the Hough transform [36], which describes a similar process but for arbitrary curves in spaced with an arbitrary amount of dimensions. The Radon transform describes a function that maps the line integral of a function f along a line g with the offset s in the direction ϑ to a position (s, ϑ) in a so-called accumulator. A visualization of the relationship between g, s and ϑ is shown in Figure 2.5.

The X-ray transform represents the mathematical expression of an X-ray experiment, where the absorption of the material influences the amount of radiation that hits the detector. The value that is measured at the detector pixel is the Radon transform in an arbitrary direction and offset s that lets the line hit the pixel. When the sample is now rotated during the experiment - which represents the function f in front of the X-ray beam - we get a linear sampling of the the semi-circle $[0, \pi)$ over all offsets that lie in the sample. Each discrete rotation angle yields one image, or slice. In Figure 2.6 the result of a Radon transform on the Shepp-Logan phantom [37] can be seen. Because points in the source image produce sinusoidal shapes in the Radon transform, it can also be referred to as sinogram.

2.4 Tomographic Reconstruction

Using the inverse Radon transform $\overline{\mathcal{R}}$, these sinogram slices can be reconstructed into the volumetric density distribution of the sample that attenuated the X-ray beam. For this, the backtransform $\mathcal{B}$ is defined at first

$$\mathcal{B}f(\boldsymbol{x}) = \int_{\vartheta=0}^{\pi} f(x_1 \cos\vartheta + x_2 \sin\vartheta, \vartheta) d\varphi, \qquad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2. \tag{2.13}$$

Unfortunately this is not the same as the wanted back transform $\overline{\mathcal{R}}f(\boldsymbol{x})$. Therefore the Fourier slice theorem is used to do the back projection.

Theorem 1 (Fourier Slice Theorem) *The one-dimensional Fourier transform of a parallel projection is equal to a slice through the middle of the two-dimensional Fourier transform of the original object that was projected. Knowing the one-dimensional Fourier transforms of all the projections, it is therefore possible to reconstruct the two-dimensional Fourier transform of the original object. [38]*

Essentially this means applying a one dimensional Fourier transform along the direction defined by the angle θ at the perpendicular offset σ.

$$\mathcal{F}\mathcal{R}f(\sigma, \theta) = \widetilde{f}(\sigma\cos\theta, \sigma\sin\theta), \qquad \sigma \in \mathbb{R}, \theta \in [0, \pi). \tag{2.14}$$

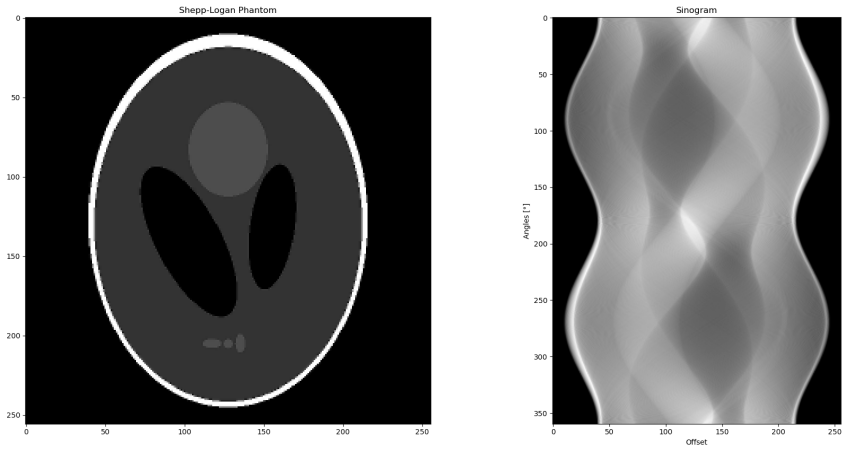


FIGURE 2.6: Shepp-Logan phantom and corresponding sinogram over the full circle with 360 projections.

Then a two dimensional inverse Fourier transform is applied to the result, while using a conversion between polar and Cartesian coordinates

$$\boldsymbol{\omega} = \begin{pmatrix} \omega_1 = \sigma\cos\theta \\ \omega_2 = \sigma\sin\theta \end{pmatrix} \tag{2.15}$$

to account for the rotation of the sample.

$$f(x) = \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \widetilde{f}(\omega)\mathrm{e}^{-2\pi\mathrm{i}x\omega}d\omega. \tag{2.16}$$

Changing the integration limits and applying equation (2.16) this can be rewritten as

$$f(x) = \frac{1}{2\pi}\mathcal{B}\overline{\mathcal{F}}_\sigma(|\cdot| \star \mathcal{F}_s\mathcal{R}f)(x), \tag{2.17}$$

where $\star$ denotes pointwise multiplication. Using the convolution theorem above equation can be once again rewritten to give the formula for the filtered back-projection

$$f(\boldsymbol{x}) = \frac{1}{2\pi}\mathcal{B}\big((\mathcal{R}f) * g\big)(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^2, \tag{2.18}$$

where

$$g(s) = \int_{-\infty}^{\infty} |\sigma|\mathrm{e}^{-2\pi\mathrm{i}s\sigma}d\sigma, \qquad s \in \mathbb{R}. \tag{2.19}$$

Theorem 2 (Convolution Theorem) *An important property of the Fourier transform is that a convolution operation $f * g$ in real space can be represented by a pointwise multiplication, denoted by the $\star$-operator, in Fourier space [6]. That is*

$$f * g = \overline{\mathcal{F}}(\mathcal{F}(f) \star \mathcal{F}(g))). \tag{2.20}$$

With (2.18) the slice images taken in the X-ray tomography experiments can be transformed to give a volumetric model of the density distribution in the object, depending on the attenuation of the radiation.

2.4.1 Analytic Reconstruction of Cone Beam Data

The technique described above is used to reconstruct tomographic data. It is called filtered backprojection (FBP), because the oversampling in the lower

frequencies and the undersampling in the higher frequencies are compensated using a filter. The mathematically correct way to do this, is to apply the ramp filter, as described in (2.19)) [38].

A limiting factor of FBP is that it still assumes the presence of perfectly parallel light paths, often called parallel beam geometry. Laboratory sources such as the FlexCT system in the VisionLab [8] use X-ray sources that do not provide such a parallel beam. Instead the distance between source and detector is small. This, in combination with the shape of the anode, causes the beam shape to resemble a cone instead, naming the resulting geometry a cone-beam geometry. This geometry is the most prevalent in laboratory sources, and so a reconstruction technique that does not make wrong assumptions is required.

The Feldkamp, Davis, and Kress (FDK) algorithm is an approximate theoretical solution to this problem [39]. Its base principle is to re-sample projection images acquired with a known and precisely defined cone beam setup into parallel slices of fan beam projections and then solving the tomography problem with a modified Fourier Slice Theorem. This modified version of the backprojection was first described by Parker [40].

Visualizations of the cone beam, fan beam and parallel beam geometries are shown in Figure 2.7. Fan beam data, as shown in Figure 2.7 (b), is a series of projections acquired with a line detector, where the light source distributes light in a triangular area, the outermost rays hitting the detector after longer traversal time than the central rays. The name fan beam derives from its shape. Its equivalent in 3D is the cone beam, where the triangular cross section is apparent both in the vertical as well as the horizontal axis of the detector.

The reconstruction in the fan beam case uses a weighting scheme to properly distribute the intensity values with respect to the locally changing geometry.

$$\mathcal{B}_{\mathrm{FDK}} f(\boldsymbol{x}) = \int_{\vartheta=0}^{\pi} w_{\mathrm{fan}} f(x_1 \cos\vartheta + x_2 \sin\vartheta, \vartheta) * g \, \mathrm{d}\vartheta \tag{2.21}$$

where

$$w_{\mathrm{fan}} = \frac{R^2}{(x_1 \cos\vartheta + x_2 \sin\vartheta)^2} \tag{2.22}$$

is the weighting to approximate fan beam data from the cone beam data. The parameter R is the distance to the detector. The filtering term g is the same as defined in (2.19). The filter is applied per row instead of the total image.

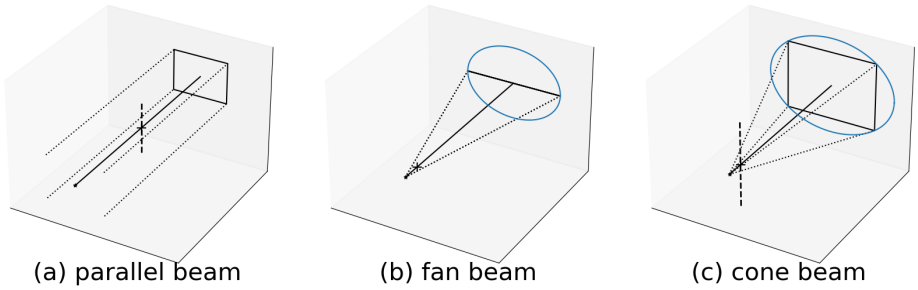


FIGURE 2.7: Schematics of the basic projection geometries parallel beam (a), fan beam (b), and cone beam (c). Indicated as a star is the source position, a cross denotes the detector center and the lines show the detector extents as well as the area covered by the X-ray beam. Shown superimposed in blue on the detector is a circular region of the smallest cone that contains the detector fully.

This filtering is crucial step in this reconstruction method. The backprojection process ”smears” the recorded intensity over the reconstruction area. Due to the concentric circular geometry, the sampling of the intensities is denser in the center of the circular reconstruction region and less dense towards the outside. The filtering operation weighs the intensities inversely proportional to the sampling density, generating a less blurry reconstruction. An illustration of the effect of the filtering is shown in Figure 2.8.

2.4.2 Algebraic Reconstruction

Another way of reconstructing an object from its projections is algebraic reconstruction. This method of reconstructing formulates the problem of obtaining a reconstructed image as a set of mathematical equations. The light intensities measured on every single pixel on the detector are seen as independent pieces of information which reflect the solution of a line integral from the source to that pixel. Expressing the line integral of each projected pixel as an linear equation with a known solution, the operation of projection can be expressed in the so-called projection matrix $\boldsymbol{W}$

$$\boldsymbol{p} = \boldsymbol{W}\boldsymbol{x}_v, \tag{2.23}$$

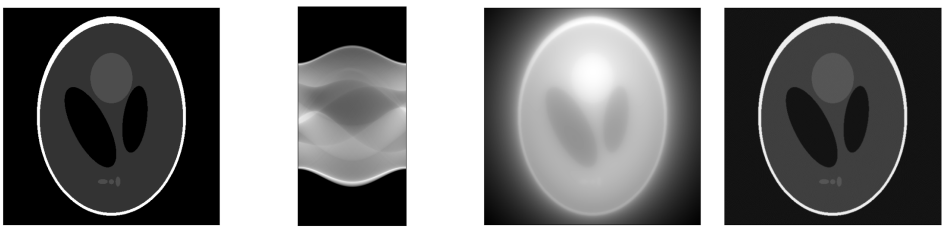


FIGURE 2.8: Effect of filtering in the filtered backprojection. Shown is the Shepp-Logan phantom (a), 360 equiangular projections (b), the unfiltered reconstruction using backprojection (c) and the filtered reconstruction (d).

where $\boldsymbol{p}$ are the obtained projections and $\boldsymbol{x}_v$ is the object/volume that caused the projections. Both $\boldsymbol{x}_v$ and $\boldsymbol{p}$ are considered vectors, the projection matrix $\boldsymbol{W} \in \mathbb{R}^{NxM}$, where N is the number of elements in the reconstruction and M the number of elements in the projection. As a system of linear equations, this matrix also has an inverse. This inverse can be used to compute the original object $\boldsymbol{x}_v$ from the projections $\boldsymbol{p}$ if the projection matrix is known and invertible.

$$\boldsymbol{x}_v = \boldsymbol{W}^T \boldsymbol{p}. \tag{2.24}$$

The FDK algorithm would be expressed as

$$FDK(\boldsymbol{x}_v) = \boldsymbol{W}^{\mathrm{T}}(\boldsymbol{g} * w_{\mathrm{fan}}(\boldsymbol{x}_v)) \tag{2.25}$$

in this notation [41].

The term "algebraic reconstruction" is just another name for solving the tomography problem using iterative solvers for large linear systems of equations. One very popular such reconstruction algorithm is called simulataneous iterative reconstruction technique (SIRT), which is a technique based on gradient descent.

With gradient descent, a function's minimum (in this case the reconstruction) is iteratively found by searching a minimum along a line and from there

computing a new line and searching the minimum again until a global minimum is found, or another stop condition is met. Using the notation analogous to [42], the line search can be formulated as a one-dimensional optimization problem:

$$\min_{\alpha>0} f(x_k + \alpha p_k). \tag{2.26}$$

The direction of this line search is the direction of the steepest descent, which mathematically is given by the gradient of f

$$p_k = -\nabla f(x_k) \tag{2.27}$$

The subscript k indicates the guess at the k'th iteration of the optimization. A new guess $x_{k+1} \in \mathbb{R}$ is given based on the previous iteration

$$x_{k+1} = x_k + \alpha p_k \tag{2.28}$$
$$= x_k - \alpha \nabla f(x_k). \tag{2.29}$$

Written in matrix form as the FDK algorithm above, the gradient descent algorithm becomes

$$\boldsymbol{x}_{v,k+1} = \boldsymbol{x}_{v,k} - \alpha \nabla f_{\text{loss}}(\boldsymbol{x}_{v,k}) \tag{2.30}$$
$$\boldsymbol{x}_{v,k+1} = \boldsymbol{x}_{v,k} - \alpha \boldsymbol{W}^T(\boldsymbol{W}\boldsymbol{x}_{v,k} - \boldsymbol{p}_k), \tag{2.31}$$

where the projection difference

$$\boldsymbol{W}\boldsymbol{x}_v - \boldsymbol{p} \tag{2.32}$$

is chosen as the objective function f_{loss} for the minimization of the problem [43]. The step size α along the gradient direction can be varied.

The SIRT algorithm is a variation of this algorithm and based on a class of solvers which include other well known methods like "Cimmino's method" and "Kaczmarz's method" [44]. SIRT updates all elements in the reconstruction in one single iteration, hence the "simultaneous" in its name. The initial estimate is often just a vector of zeros and is iteratively updated using the residual of the projection difference, weighted with the inverse column and row sums, represented by two diagonal matrices $\boldsymbol{C}_1$ and $\boldsymbol{R}_1$, respectively. An overview of the algorithm is given in Algorithm 1.

The similarity of line 5 in Algorithm 1 to Equation 2.31 is apparent. Here the diagonal matrices $\boldsymbol{C}_1$ and $\boldsymbol{R}_1$ function as both the step size α as well as

Algorithm 1 SIRT

$\boldsymbol{x}_{v,0} \leftarrow \boldsymbol{x}_v$
$\boldsymbol{C}_1 \leftarrow diag\left(1/||c_j||_1\right)$
$\boldsymbol{R}_1 \leftarrow diag\left(||r_i||_1\right)$
for k = 0; k < num_iter; k++ **do**
    $\boldsymbol{x}_{v,k+1} \leftarrow \boldsymbol{x}_{v,k} + \boldsymbol{C}_1\boldsymbol{W}^T\boldsymbol{R}_1(\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}_{v,k})$
end for

a regularization factor based on the sums of the contributing pixels in the reconstruction $\boldsymbol{x}_v$ and projection $\boldsymbol{p}$.

In the same way, other, more general purpose algorithms like Newton-Krylov optimizers [45] or the Barzilai-Borwein (BB) method [46], can be used to solve the tomography problem. The BB method is a variation of the gradient descent algorithm, where the parameter α is chosen to approximate the ideal step size for steepest descent and is used in the presented research for reconstructions in addition to SIRT. The way to compute that step size is shown in Algorithm 2. The dot product of the differences between the current best estimate $\boldsymbol{x}_{v,k}$ and the previous best estimate $\boldsymbol{x}_{v,k-1}$ and the differences between the gradient of the current iteration and the previous iteration is divided by the magnitude of the gradient differences.

Algorithm 2 BB

$\boldsymbol{x}_{v,0} \leftarrow \boldsymbol{x}_v$
$\texttt{grad} \leftarrow \boldsymbol{W}^T(\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}_{v,k})$
$\alpha \leftarrow 1/||\texttt{grad}||$ ▷ initial step size
for k = 0; k < num_iter; k++ **do**
    $\texttt{gradp} \leftarrow \texttt{grad}$ ▷ Store gradient of previous iteration
    $\texttt{grad} \leftarrow \boldsymbol{W}^T(\boldsymbol{p} - \boldsymbol{W}\boldsymbol{x}_{v,k})$
    $\alpha \leftarrow ((\boldsymbol{x}_{v,k} - \boldsymbol{x}_{v,k-1}) \cdot (\texttt{grad} - \texttt{gradp}))/||(\texttt{grad} - \texttt{gradp})||$
    $\boldsymbol{x}_{v,k+1} \leftarrow \boldsymbol{x}_{v,k} + \alpha\texttt{grad}$
end for

This leads to small step sizes when the gradient magnitudes are far apart between iterations and large step sizes when they remain similar.

Both analytic and algebraic reconstruction have their uses. FDK is commonly used as a reconstruction algorithm for medical CT, where the geometry is fixed and reconstructions need to be available quickly. Algebraic reconstruction techniques usually take longer to reconstruct a usable image,

but offer more flexibility in the acquisition geometry. Some examples of such geometries are explained in the following section.

2.4.3 Non-standard geometries and the ASTRA toolbox

An advantage of algebraic reconstruction over analytic reconstruction is the fact that the system of equations does not imply any sort of physical setup in its model. Both FBP and FDK rely on the fact that the geometry of the acquisition is configured in a certain way. If the actual geometry deviates from these assumptions, the reconstruction quality can suffer or it can even be entirely impossible to reconstruct any useful data. With an algebraic approach, each pixel in the projection is independent of all the others from the viewpoint of the acquisition. This means that, at least theoretically, each of those pixel values could be measured under a different view angle or source-detector-distance, as long as the geometry is fully defined and can be related back to the object if combined with the remaining projections. Using this fact, the flexibility of laboratory sources grows. The FleXCT project at the University of Antwerp for example, uses a modified Tescan UniTOMXL scanner with 10 degrees of freedom, allowing for a set of non-standard geometries to be used during XCT scans [8]. Some possible scanning non-standard geometries are described in the following paragraphs. They are visualized using the software package `flexraytools`, which was developed by the author in the context of that project.

The All Scale Tomographic Reconstruction Antwerp /Amsterdam (ASTRA) toolbox [47] is a software toolbox for algebraic and analytic reconstruction developed at the University of Antwerp in collaboration with the Centrum voor Wiskunde en Informatica (CWI) in Amsterdam. The toolbox provides a Python and MATLAB interface to a number of C++ routines that, with the support of graphical processing units (GPUs), can simulate mathematically correct forward projections and reconstruct volumetric data from acquired projections. The interface is designed to allow for arbitrary configurations of parallel or cone beam geometries.

A common problem in clinical settings is that patients breathe and move slightly during a scan. The standard "step and shoot" method often causes misalignment between acquisitions of different projections, making the image blurry and in the worst case unusable [48]. To speed up the acquisition and reduce this source of errors, the source and detector are rotated continuously while the patient is moved through them. This results in a helical

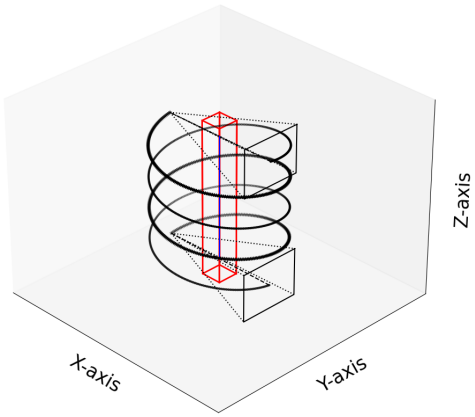


FIGURE 2.9: Trajectory of source ($*$) and detector ($\cdot$) during a helical scan. The blue line indicates the linear movement axis, here the Z-axis. The red box indicates a possible volume that can be imaged by this geometry. Of course, the direction for a medical scan would be rotated by 90 °, rotating around a bed.

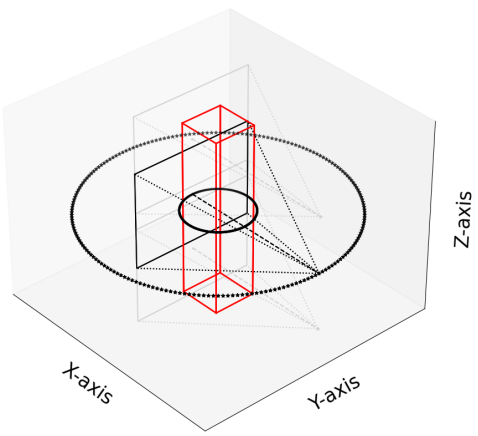


FIGURE 2.10: Trajectory of source ($*$) and detector ($\cdot$) for a standard cone beam scan and a source and detector pair stacked on the top and bottom, showing the geometry of a stacked scan with a ”step and shoot” acquisition scheme. The geometry has been exaggerated for visualization purposes.

trajectory of source and detector and makes it possible to reconstruct the region of interest (ROI) in a single continuous acquisition, usually short enough for the patient to hold their breath. Helical CT can also be applied in material science, where objects that do not fit into the field of view (FOV) can be scanned fully using a helical acquisition scheme, as shown in Figure 2.9.

For better reconstruction quality the traditional "step and shoot" method can also be combined with a linear displacement along the rotation axis, also increasing the field of view (FOV) in the direction of the linear displacement, as shown in Figure 2.10. To ensure the reconstruction to be possible, the scans are made with some overlap, so that the images can later be registered to one another. They are then converted to one continuous projection image to be reconstructed.

As long as the acquisition can be performed fast enough, an acquisition can also be performed as a time resolved acquisition. This is referred to as 4-dimensional (i.e. time-resolved) computed tomography (4D CT). During a typical scan with a standard rotating geometry, the acquisition is performed several times and several reconstructions are made. Each individual reconstruction represents a time snapshot of the state of the observed object at that time. Using motion correction and other techniques using temporal information, the reconstructions can be improved [49].

Variation of the acquisition geometry is heavily utilized in NDT. The time resolved acquisition of tomographic images is a great tool to investigate compression properties of materials, reaction of a material to extreme temperature or the analysis fluid dynamics. In cases where the investigated sample is limiting the geometry of the scan, algebraic reconstruction techniques can still reconstruct the structure from limited angles [50] and the use of additional information aids in improving the data quality in these cases [51]. It is also possible to reconstruct smaller parts of large samples using so called ROI tomography [52].

2.5 Investigation of Material Structures

The work presented in this thesis focuses on the analysis of composite materials and the analysis of their components using non-destructive testing. This chapter introduces the production of said materials, their properties and how they are analyzed in the industry.

Material structures come in a wide variety of shapes and forms. Traditionally in material engineering and material science the structure of a material

becomes important when the material currently used for a task is failing to do this task well enough or for long enough. This can be a mechanical part failing because of repeated cyclic loads or exposure to forces that wear it down over time [53]. Other use cases can be found in the search for a completely new material that is supposed to replace another with a cheaper, lighter, stronger or otherwise more advantageous material. One class of highly researched materials are so-called compound materials. Those materials combine good properties of multiple materials to make a better, unified material. An example for such a material is the subject of this thesis - a glass fiber-reinforced polymer (GFRP). GFRP combine a polymer resin of some chosen polymer as the material matrix and reinforce the structure by adding fibrous material, in this case glass fibers. The polymers are easy to work with, as they are easily moldable and can cure to very hard materials. However the material in itself can be very heavy and has low tensile strength when it becomes too thin. Adding fibers increases the strength of the material body, allowing it to resist much greater forces than without the added fibers [54]. They also increase the strength-to-weight ratio, which enables the use of less material for the same strength in a component [2].

One subject of material science is thus the analysis of what kind of materials or which combination of materials gives the most advantageous compound for a given task. A valuable tool in this research is the non-destructive testing of samples of those materials.

2.5.1 Glass fiber-reinforced polymers

Glass fiber-reinforced polymers are polymer materials whose mechanical properties have been altered by introducing glass fibers into the material during the manufacturing process.

To produce the glass fibers, glass is melted in furnaces at 1310 - 1390 °C, extruded through a heated feeder channel and funneled into a so-called bushing containing anywhere between 400 and 1600 or more small holes of around 2 mm diameter. The bushing forms the molten glass into thin streams, which can be pulled by the already cooled section with a constant force and speed of up to 80 m/s. Due to the pulling the cross section diameter reduces drastically to approximately 10 µm, like shown in Figure 2.11. Once the flow of glass is established, the fibers can be continuously produced and rolled up on a spindle, named the "spun cake" or directly processed further into a woven material or small chopped pieces [55]. They are also often coated with a material that prevents breaking.

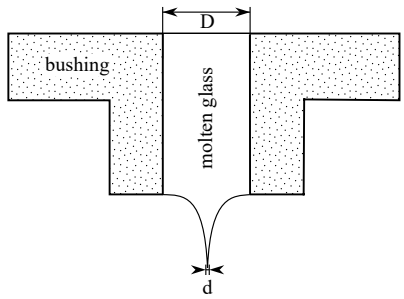


FIGURE 2.11: Principle of glass extrusion from a single hole in the bushing. The input hole diameter D narrows down to the filament diameter d, which is usually around 200 times smaller. This figure is a simplified recreation of Fig. 22 in [55].

These glass fibers are added into a liquid polymer resin and can be injection molded or cast into various shapes. The building of suitable molds and the design and modeling of the injection molding process is a complex topic requiring various fluid and flow simulations [56]. During the injection molding process, the polymer is heated to liquidity and then injected into a mold as a fiber suspension. The fibers are mixed in just before being pumped out of the nozzle [57]. Fibers usually align along the flow direction of the matrix material close to the injection point and deviate in orientation due to turbulent flow or the restrictions of the mold boundaries. Usually the fiber orientation is different at the mold border compared to the body of the resin. Even the polymer without fibers in it aligns itself because of the shearing and stretching that happens to the molecule chains in the flow [56]. The amount of fibers, their distribution, length and thickness as well as flow rate, polymer viscosity and cooling curve are important parameters in this process that heavily influence the final product [56]. Because of this vast amount of parameters, simulations are essential to predict the outcome of the manufacturing process, but testing after a part is manufactured remains crucial as well. The design process of GFRP is thus similar to an iterative optimization with simulation, manufacturing and NDT as the elements of each iteration. Models are used in this context to simulate NDT experiments with material parameters that address shortcomings of the previous iteration, which are then verified after the manufacturing of the new material.

Of course the fibers can also be mixed into the resin without injection, similar to what happens with fiber reinforced concrete. The fiber orientation distributions in such a material are random and can be almost isotropic [58].

Other manufacturing methods use weaves of fibers that are laid into a resin to reinforce it or even applied to a surface to reinforce and the polymer is brushed on like paper-maché.

Chapter 3

Parametric Reconstruction of Straight Fibers

As described in the previous chapter, non-destructive testing plays a big role in the research of materials. This thesis describes the efforts in a particular sub-topic of non-destructive testing - fiber-reinforced composite analysis. Fiber reinforced composites, while strong and versatile, are still being developed and new combinations of material matrices with different types of fibers are being analyzed. The work done in this thesis aims to provide a set of tools in addition to the existing frameworks, that lets researchers analyze fiber composites on an individual fiber level and conduct further investigations from there.

Before any analysis can take place, a sample of, for example, a GFRP has to be analyzed in a tomographic scanner and a reconstruction of the sample is computed.

Fiber analysis methods and algorithms that work on those reconstructions all have common steps that have to be taken. At first the voxels in the reconstruction have to be classified into "fiber" and "non-fiber" voxels. With all the "fiber voxels" in the reconstruction identified, the methods start diverging. Some methods extract statistical information directly from the volume [59]. Other methods analyze fill percentages [60] and yet other methods start clustering the voxels to obtain representations of the individual fibers, which in turn can be used to obtain statistical and geometrical information about, i.e. the orientation distribution [61], length distribution [62] and/or models of the entire fiber population [63]. It is also possible to model individual fibers. In that case, it is important to segment those fibers, which can be done using traditional image processing [64] or neural network based methods [65].

The contribution(s) discussed in this thesis fall into the last category. In

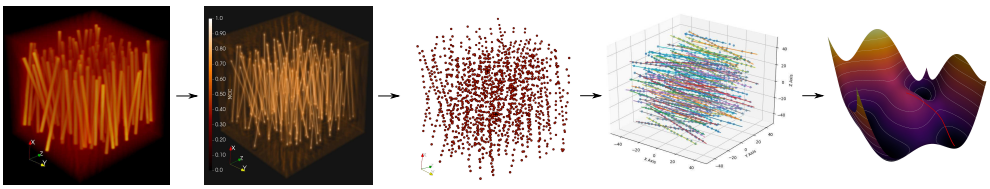


FIGURE 3.1: Processing pipeline of the fiber analysis method discussed in this chapter. Starting from a reconstruction, the voxels representing fibers are identified, clustered and converted into mathematical representations and then optimized.

the following sections, the steps mentioned above will be explained in general, followed by a detailed description of the methods used in the contributions. An overview of these steps is show in Figure 3.1.

3.1 Fiber Detection

What differentiates a fiber voxel from a non-fiber voxel? In many image processing tasks - be it 2D pixels or 3D voxels - the first question that arises will be the separation or segmentation of an object or classes of objects from some kind of background. The simplest way of doing this is a so-called "thresholding" operation. Given a threshold value that lies somewhere in the interval of the pixel values, this operation assigns a low value, often 0, to any pixel whose value is lower than that threshold, and a high value, often 1, to any other pixel. Methods like Otsu's thresholding [66] are designed to separate images containing exactly two classes by using the information about the pixel value distribution contained in the histogram.

This very simple operation does of course not always lead to the proper results. Often the intensity distribution in an image is too in-homogeneous to make use of a static threshold. Methods like local adaptive thresholding [67] can remedy some of those problems, but often times more sophisticated processing chains are required.

The tomographic images of fibers that are dealt with in this thesis suffer from intensity inhomogeneity due to reconstruction, physical effects like beam-hardening [6] and sometimes inadequate resolution.

Current methods to characterize the structural properties of GFRP from high resolution µCT images rely on a sequential work flow comprised of volumetric reconstruction from a large number of projections (typically > 1000)

and subsequent fiber segmentation and image analysis [68, 69, 70]. Many of them involve extracting the center lines of the fibers. However, their individual methods of extracting the center lines and the use of the data differs in these approaches. Emerson et al [71] use a dictionary learning approach to extract the centers of very low resolution fibers slice by slice, relying on the unidirectional fiber direction distribution of their datasets. Pinter et al.[62] use the local Eigenvalues and a circular voting approach. Huang et al.[72] use skeletonization to extract the center lines and some use a template matching based approach [73]. Other methods involve fitting cylinder segments to local orientation vector fields [74] to detect the fibers by "walking" along the center line. Of course, deep learning can also be employed for this purpose [65]. In the method presented in the following chapter, the fiber extraction is based on template matching.

3.1.1 Template Matching

A common task in image processing is to find an object in an image, or in this case a volume. If the shape of the object is known, an approach called template matching can be employed to solve this task.

This can be done using something like the morphological hit-or-miss transform, or a cross correlation, also commonly referred to as template matching. A cross correlation of a function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ and a template $g : \mathbb{R}^2 \mapsto \mathbb{R}$ is defined in terms of a convolution of f with g^*, a mirrored version of g [75]

$$(f * g^*)(x) = \int_{\mathbb{R}^2} f(y)g(y-x)dy, \quad x \in \mathbb{R}^2. \tag{3.1}$$

This can be trivially extended to $\mathbb{R}^3$. From (3.1) it can be seen that the template g is essentially moved over all locations of f and the correlation is recorded in a second image or volume at that location. Note that this is not the same as the statistical measure called correlation. The smaller the difference between the template and the local region it covers, the higher the correlation value in that location. The template can then be found in the image by detecting local maxima, as shown in Figure 3.2.

Lewis [76] describes a fast variant of this procedure that additionally normalizes the intensities, such that only the "shape" of the template is matched. It is called normalized cross-correlation (NCC). It allows finding templates independently of the contrast or absolute intensity, as long as the gray values

match. Using the integrals of the image and template

$$\bar{f} = \int_{\mathbb{R}^2} f(x)dx$$
$$\bar{g} = \int_{\mathbb{R}^2} g(x)dx$$

to center the value distributions and the distance of the image and template from these integrals

$$s_f = \sqrt{\int_{\mathbb{R}^2} (f(x) - \bar{f})^2 dx}$$
$$s_g = \sqrt{\int_{\mathbb{R}^2} (g(x) - \bar{g})^2 dx}$$

as a normalization factor, the NCC can be written as [75]

$$\frac{(f - \overline{f}) * (g - \overline{g})^*}{s_f s_g}. \tag{3.2}$$

3.1.2 Template Matching to extract Fibers

In a CT reconstruction, fibers have circular cross section, with the intensity falling off smoothly towards the background. To detect the voxels representing a fiber using template matching, the choice fell on a Gaussian with standard deviation that would make the full-width at half-maximum equal to the fiber radius r_{fiber}

$$\sigma = \frac{r_{\text{fiber}}}{2\sqrt{2\ln 2}}. \tag{3.3}$$

After applying this template using Equation 3.2, the local maxima are extracted from the resulting image. The voxels with an intensity lower than an empirically chosen threshold $t \in \mathbb{R}$ are set to 0, the remaining voxels are left untouched. The intensities of the non-zero voxels in the template matching result are then cross referenced with the original image. Voxels that do not

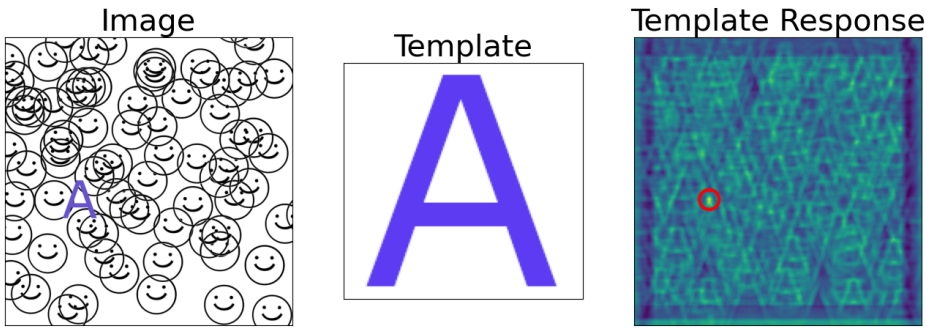


FIGURE 3.2: Image of overlapping smiley faces hiding the letter A (left), the letter itself (middle) and the response of the normalized cross correlation when using the letter as a template (right). For better visibility the letter is colored differently in the image, the template response was calculated on grayscale images. The red circle marks the location of the peak in the response.

have an intensity within 25% of the fiber attenuation μ_{fiber} are discarded. All remaining voxels are used for the following step, the clustering [77].

3.2 Straight Fiber Model

The steps described in the previous section all involve processing of the voxel volume and are therefore bound by the resolution of a voxel. While this representation is straightforward and sufficient for a lot of use cases, it does have several disadvantages in the case of fibers. Fibers are long, thin, stick-like objects that lie in arbitrary orientations inside an embedding polymer. The resolution of a single voxel must be significantly smaller than the fiber diameter to be able to see and detect the fibers in a reconstruction. The minimum amount of pixels to detect a feature in an image is given by the Nyquist-Shannon sampling theorem [78], which implies that the smallest feature detectable in a discrete image is sampled by 2 pixels. It is, however, better to have more data available for a robust detection. A rule of thumb in image processing is that an object can be effectively detected when its features are spread over 5 pixels. Following that rule of thumb and assuming a glass fiber diameter

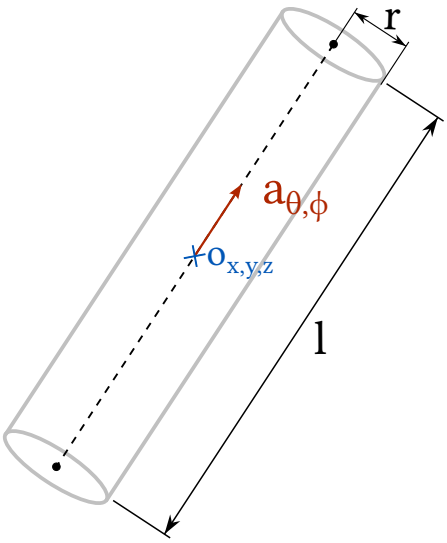


FIGURE 3.3: Cylinder model of a fiber with the parameters used for describing the fiber indicated. Note that the radius r is assumed to be constant and does not need to be determined.

of 14 µm the voxel resolution should be in the order of 2 µm for reliable detection. This also means that with a length of around 400 µm for glass fibers, 200 voxels are needed to get an accurate image of the entire fiber. If the center line of the fiber is used to represent it as an "object" in some analysis algorithm, hundreds of coordinates are needed to describe the individual fiber. With a large number of fibers this can quickly get inefficient. It is therefore convenient to describe the entire fiber object by a concise set of parameters, which not only are more compact than chains of voxel coordinates, but also suffer less from stepping artifacts from the discretization on the lattice grid.

Straight fibers can be conveniently described as cylinders with a very high aspect ratio. Assuming the radius of the fiber is known, this means a fiber can be completely described by a center position, a direction vector and a length. Using spherical coordinates to describe the direction unit vector, this results in a descriptor based on 6 scalars.

$$o_x, o_y, o_z, a_\theta, a_\phi, l.$$

The center of the fiber is represented by $o_x, o_y, o_z \in \mathbb{R}$, the orientation by $a_\theta \in [0, 2\pi)$ and $a_\phi \in [0, \pi)$, and the length by $l \in \mathbb{R}$. A visualization of the

parameters is shown in Figure 3.3.

3.3 Fiber Clustering

In the fiber detection algorithm used here, it is assumed that fibers are straight, cylinder-like objects of higher density inside the volume. As highlighted in Subsection 2.3.5, points or small circular objects convert to sinusoidal curves in the projection space. This gives the set of projection images of a scan the name sinogram. In the same transformation, a straight line in real space maps to a point in projection space. This fact can be used to detect line-like structures inside images.

3.3.1 Hough Transform

A generalization of the Radon transform is the so-called Hough transform, which was already discussed in Subsection 2.3.5. It describes an integral along a generalized curve C.

$$\int_C f(x)\mathrm{dx}. \tag{3.4}$$

It is required that C is a parameterized curve. It can be described as the set $C = \{x(t) : t \in D_x\}$, with x a differentiable, parametric function of t in the domain $D_x \in \mathbb{R}$. After using the substitution $\mathrm{d}x = ||x'||\mathrm{d}t$ this becomes

$$\int_{D_x} f\left(x(t)\right) ||x'(t)||\mathrm{d}t. \tag{3.5}$$

Assuming the parameters of C are represented by a parameter vector $(a_1, a_2, \ldots, a_n)$, the general Hough transform is given by

$$\hat{f}(a_1, a_2, \ldots, a_n) = \int_{D_x} f\left(x(t)\right) ||x'(t)||\mathrm{d}t, \quad (a_1, a_2, \ldots, a_n) \in \mathbb{H}, \tag{3.6}$$

with $\mathbb{H} \subseteq \mathbb{R}^n$ the Hough parameter space or accumulator [75].

As fibers can be seen as straight line segments with some thickness, consider the parametric line description introduced in Subsection 2.3.5. A line can be represented by an angle ϑ and some offset $\boldsymbol{s}$ from the origin. This representation is used in the Radon integral, as it gives a convenient description of the line integrals in real space, coding the projection space as a parameter

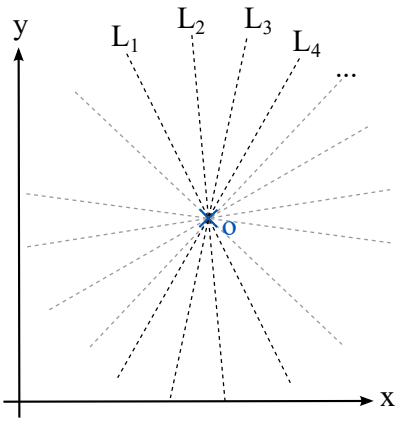


FIGURE 3.4: Lines passing through a common point p. The point o will have a high value in the Hough accumulator, as all the integrals of the lines passing through it get added at its position. This figure is a reproduction of Figure 3.4 from [79] with slight changes.

space spanned by the angle ϑ and the perpendicular distance of the offset from the origin. Extending the representation from two to three dimensions, a line can be represented by a position vector in 3D space, the offset from the origin, and two angles for the (unit) direction of the line in spherical coordinates.

Consider a point o in space and all lines L_n passing through that point as shown in Figure 3.4. If all the points on each line would be assigned the value of the integrals of all lines passing through them, there would be a local maximum in that point. The Hough transform operates on this principle. An accumulator array with the dimensionality of the curve parameter space is created with some discretization. Each cell in that discretized accumulator will be assigned the sum of all line integrals passing through them. Thus a local maximum in the accumulator indicates the presence of the parameterized shape with the parameters being the coordinates of the center of the cell [79].

A variation of the Hough transform, called the iterative Hough transform [80] is used in the Parametric Reconstruction (PARE) algorithm. It works on scattered point data and changes the accumulator parameters. As described above, such a vector form requires 2 or 3 parameters for the direction vector (depending on the representation) and 3 parameters for the position vector, which in the end would require an accumulator of 5 or 6 dimensions. In the

algorithm used here, the lines are described with 4 parameters [81] in a form called Robert's parameterization [82], saving on the number of dimensions and thus storage and computaional resources. Roberts represents the line with the parameters x_0, y_0, ϕ, θ. The line direction vector is represented in spherical coordinates, (ϕ, θ), omitting the radius $r = 1$ for the unit vector, which can be used to calculate the direction vector in Cartesian coordinates

$$\boldsymbol{b} = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} \cos\phi\cos\theta \\ \sin\phi\cos\theta \\ \sin\theta \end{pmatrix}. \tag{3.7}$$

The azimuth $\theta \in [0, 2\pi)$ and the elevation angle $\phi \in [0, \pi)$. The position $\boldsymbol{b} \in \mathbb{R}^3$.

The position of the line is defined by an arbitrary point in space, but contains redundant information. Thus, Roberts computes the two coordinates x_0, y_0 using the information contained in the direction vector and the offset vector $\boldsymbol{o} = (o_x, o_y, o_z)$ of the line

$$x_0 = \left(1 - \frac{b_x^2}{1 + b_z}\right) o_x - \left(\frac{b_x b_y}{1 + b_z}\right) o_y - b_x o_z \tag{3.8}$$

$$y_0 = \left(\frac{b_x b_y}{1 + b_z}\right) o_x - \left(1 - \frac{b_y^2}{1 + b_z}\right) o_y - b_y o_z. \tag{3.9}$$

The angular accuracy of the iterative Hough Transform algorithm is limited by the sampling of directions and relies on the iterative subdivision of an icosahedron's faces into triangles as visualized in Figure 3.5. Each vertex of the subdivided icosahedron is projected onto the unit sphere to yield the unit vector of the sampled direction.

The spatial discretization depends on the size of the axis-aligned bounding box (AABB) of the points the transform will be applied to. The AABB is described by two points $\boldsymbol{o}_{\min} = (x_{\min}, y_{\min}, z_{\min})$ and $\boldsymbol{o}_{\max} = (x_{\max}, y_{\max}, z_{\max})$, which are the minimum and maximum of the coordinates in the points that the transform will be applied to. The distance between two samples d_{sample} is computed as

$$d_{\text{sample}} = \frac{\sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2 + (z_{\max} - z_{\min})^2}}{64}$$

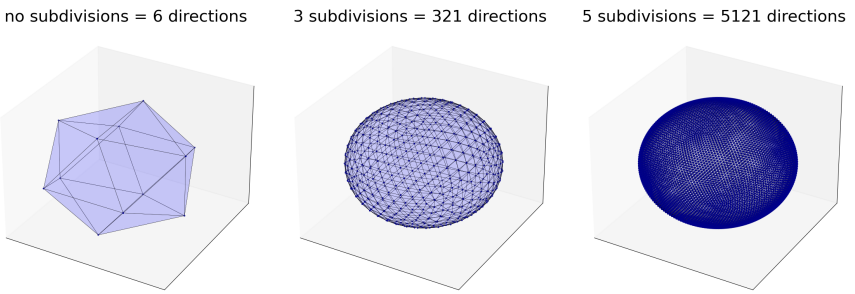


FIGURE 3.5: Several subdivisions of an icosahedron to produce uniformly distributed directions on the unitsphere. The indicated number of unique directions is half the number of vertices, as the polyhedron is mirrored along the x-y plane passing through the origin. The initial icosahedron has 12 vertices, giving 6 unique directions.

from the diagonal of the AABB. The number 64 is arbitrarily taken from the original source code that the authors of [80] provide.

3.3.2 Assigning Fiber Voxels to Clusters

Using the iterative Hough transform [80], the fiber voxels detected as described in Section 3.1 can be combined into individual fiber clusters. The iterative Hough transform algorithm initially performs a normal Hough transform on the point cloud and determines the parameters of one line using the single highest peak in the accumulator. In the next step, the algorithm finds all points in the point cloud that are within a distance of dx to the found line.

With the points close to the line identified, a least squares fit is performed to yield the parametric description of the line. The points used in this process are removed from the point cloud. Then this process is repeated until either all points in the point cloud are removed or there are less than the minimum amount of points for a line fit left. The minimum amount of points for the line fit is a user parameter and is set to $n_{\text{min}} = 3$.

As a trade off between accuracy and speed, 5 subdivision steps are chosen for this use case. Subdividing the icosahedron 5 times results in 5121 samples with an average spacing of 2 $^\circ$ [81].

After detection in the Hough accumulator the lines are described in the traditional form using a 3D position vector and a vector with spherical coordinates for the direction. As the radius of the fiber is assumed to be constant and known *a priori*, the only remaining parameter is the length of the fiber, which is of course not contained in the line equation.

Due to the fact that the iterative Hough transform deals with point clouds, an initial estimate of the length can be obtained by the distance of the two points furthest from the line's centroid. Naturally, only points used in the least square fit are considered for this computation. The maxima detected in the template matching step (see Subsection 3.1.1) can be offset from the true fiber end by a length in the order of the fiber radius, because the maximum value of the template matching always lies in the center of the template, here a Gaussian. The additional thresholding step can move the point by up to a voxel. Therefore the fiber length is systematically underestimated.

To remedy this, the fiber ends need to be detected. Using Bresenham's line algorithm [83] the voxel values along the fiber axis are recorded, with the line segment symmetrically extending 0.75 times the length of the fiber on either side of the centroid. The resulting 1-dimensional intensity profile is then filtered by a median filter [84] to remove high frequency noise while preserving edges in the signal. Additionally a Gaussian filter [84] is applied to the signal, as it preserves inflection points in step functions, but effectively smoothes the signal. The filtered signal is then interpolated using B-Splines [85, 86], which enables detection of the two inflection points in the signal representing either end of the fiber. To that end, the locations of the roots in the second derivative with the highest absolute value in the first derivative of the spline, are chosen as candidates for being the end points. The two candidates of either side of the centroid are chosen as the points where the slope is higher than the threshold

$$\text{thresh} = \mu_{\text{polymer}} + 0.3(\mu_{\text{fiber}} - \mu_{\text{polymer}}).$$

Here μ_{polymer} is the average voxel intensity of the embedding polymer matrix and μ_{fiber} the average voxel intensity of the fibers in the reconstruction. These values, just as the radius, are assumed to be known beforehand. The precise value of the attenuation coefficients is not needed. A variation of up to 25% will change the outcome of the procedure only slightly. The final end point positions are determined by linearly interpolating the integer coordinates of the two voxel positions in the Bresenham line, using the fractional part of the detected coordinate. The new length of the detected fiber is the Euclidean

distance of the two points determined that way.

An illustration of such a line profile through a fiber and the interpolation with its derivatives is shown in Figure 3.6.

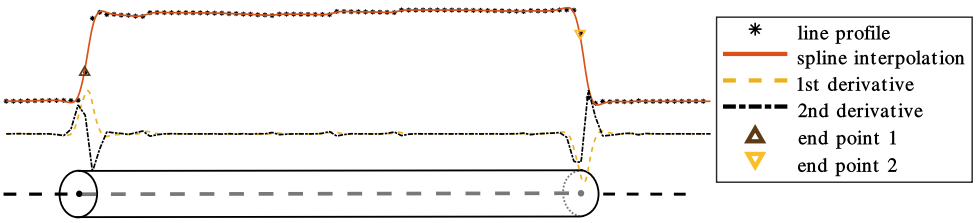


FIGURE 3.6: Line profile as it could appear in a voxel volume when sampling along a fiber, shown as a schematic below the profile. The interpolating spline (red) as well as the first (dashed orange line) and second derivatives (black dash-dotted line) and the detected end points (triangles facing up and down) are indicated. This figure is directly taken from [77], as per the guidelines of SpringerOpen. Figure reproduced from [77] under CC 4.0

3.4 Optimization of Straight Fibers

In the previous chapters the algorithms used to obtain fiber models have been described in detail. One advantage to switch to a model based representation of a fiber from a voxel based one, is that storing the information about it requires less memory than the voxel based representation. At the same time it is more accurate, as it does not suffer from discretization effects anymore. Of course, the model is obtained from a representation that does suffer from those effects, but it has the potential to have an increased accuracy. A way to achieve this, is to try and detect the fibers with sub-voxel accuracy. That is, the fiber detection algorithm determines the location and morphology of the fiber with an error smaller than the size of one voxel. Using *a priori* information about the fibers, or any object of interest that is to be detected, can result in such an increased accuracy. This is similar to the approach used in Subsection 3.3.2 to detect the end points of the fibers using spline interpolation. The *a priori* knowledge here is the fact that the material border of the fiber is, mathematically speaking, a perfect step function, as the material of the fiber instantly transitions into the material of the embedding polymer matrix. The location of the edge can therefore be determined by finding the

inflection point of the interpolating spline, which coincides with the inflection point of the step function and thus with the true location of the edge. This method uses the *a priori* information that the inflection point of a step function does not change if convolved with a Gaussian and that such a convolution can be seen as a simplified model of the point spread function [87] of the imaging system. The point spread function describes how an imaging system transforms a perfect point into some distribution, often a Gaussian. It is often expressed as the so-called optical transfer function, which describes the distribution in terms of frequency response in Fourier space [88].

Similarly, already obtained information like the projections can be used as *a priori* information. The reconstruction is obtained from the projection images and then the fiber models are detected in the reconstruction. During the processing steps information is lost. Relating the obtained model back to the projections using numerical optimization can lead to more accurate results [89].

3.4.1 Gradient Descent for Fiber Optimization

As described in Section 3.2, an individual fiber is parametrized with its center point, a direction vector and its length. The optimization of the fiber models can be described as the solution of the linear equation

$$\boldsymbol{W}\boldsymbol{x}_v(\boldsymbol{\xi}) = \boldsymbol{p}, \tag{3.10}$$

where $\boldsymbol{W}$ is the system matrix describing the forward projection operation (see Equation 2.12). The vectors $\boldsymbol{x}_v$ and $\boldsymbol{p}$ are the sample and resulting projections, respectively. Here the sample is expressed as a function of the parameter vector $\boldsymbol{\xi}$, which is a collection of the parameters of all N fibers

$$\begin{aligned}
\boldsymbol{o_x} &= (o_{x,1}, o_{x,2}, \ldots, o_{x,N}) \\
\boldsymbol{o_y} &= (o_{y,1}, o_{y,2}, \ldots, o_{y,N}) \\
\boldsymbol{o_z} &= (o_{z,1}, o_{z,2}, \ldots, o_{z,N}) \\
\boldsymbol{a_\theta} &= (a_{\theta,1}, a_{\theta,2}, \ldots, a_{\theta,N})) \\
\boldsymbol{a_\varphi} &= (a_{\varphi,1}, a_{\varphi,2}, \ldots, a_{\varphi,N})) \\
\boldsymbol{l} &= (l_1, l_2, \ldots, l_N)) \\
\boldsymbol{\xi} &= (\boldsymbol{o}_\mathrm{x}, \boldsymbol{o}_\mathrm{y}, \boldsymbol{o}_\mathrm{z}, \boldsymbol{a_\theta}, \boldsymbol{a_\varphi}, \boldsymbol{l}).
\end{aligned}$$

With this formulation the optimization/minimization problem becomes

$$\hat{\boldsymbol{\xi}} = \arg\min_{\boldsymbol{\xi}} ||\boldsymbol{W}\boldsymbol{x}_v(\boldsymbol{\xi}) - \boldsymbol{p}||^2, \tag{3.11}$$

which yields a solution vector $\hat{\boldsymbol{\xi}}$, for which the projection distance (2.32) is minimal.

This problem is solved by applying gradient descent, which is described in Subsection 2.4.2. In practical implementation, some numerical properties of gradient descent and optimization algorithms in general have to be discussed. With (3.11) we assume that the system is linear, but we do not make any assumptions about differentiability. In fact, it is not known if the problem can be formulated as a function that can be differentiated with respect to the fiber parameters. However, it is assumed that the function behaves approximately like a differentiable function and that a gradient can be approximated using a discretization technique called *finite differences* [90].

Finite differences is a way to approximate gradient information, when there is none available. If a differentiable function can be approximated by a Taylor series, its derivative can be approximated using the same series as well. Take a function $f(x)$ that is at least r times differentiable, where $r \geq 2$. Its Taylor expansion is

$$f(h) + \frac{f'(h)}{1!}(x-h) + \frac{f''(h)}{2!}(x-h)^2 + ... + \frac{f^{(r)}(h)}{r!}(x-h)^r, \tag{3.12}$$

with $h \in \mathbb{R}$, some scalar at which the function f is evaluated and $f^{(r)}$ the $r'th$ order derivative of f. Assuming the series is only continued until order 2, the equation can be rearranged to give

$$\frac{f(x-h) - f(x)}{h} \approx f'(x) + \frac{1}{2}hf''(x + \epsilon h), \tag{3.13}$$

where $0 \leq \epsilon \leq 1$. This implies that the gradient at a location $x \in \mathbb{R}$ can be approximated by subtracting a small scalar h from x and calculating the difference between the function values. The smaller the difference, the better the approximation to the real derivative.

An additional source of numerical instability is the fact that the parameters of the fibers have vastly different scales. A length difference of a couple voxels is relatively small, whereas changing the location of the fiber centroid

by the same amount can lead to a vastly different fiber model when considering the resulting projections. To reduce this problem and to be able to use a single scalar as the finite differences delta δ, the fiber parameters are scaled into the interval [0, 1].

The initial values ξ_i of a single fiber become the center of the interval, 0.5, per parameter. The boundaries of the interval, 0 and 1, are given by $\xi \pm \Delta_i$. The values for Δ_i are chosen empirically to be to be 5 voxels for the position parameters $o_{\mathrm{x,i}}, o_{\mathrm{y,i}}, o_{\mathrm{z,i}}$, three times the angular spacing in the Hough accumulator for the direction parameters $a_{\theta,i}$ and $a_{\varphi,i}$ and 10 voxels for the length l_i.

The fibers are optimized individually, keeping the remaining fibers in the population fixed. To that end, the volume $\boldsymbol{x}_v$ is pre-initialized by inserting the $N-1$ fixed fibers before starting the optimization. This is done by sampling the fibers on a regular lattice grid, "voxelizing" them. The fiber that is optimized can be inserted into this pre-existing volume with each new parameter configuration using the same technique. The projections of this generated volume are simulated using the ASTRA toolbox [47] and the projection error

$$\mathrm{RMSE}(\xi) = \frac{1}{2}\sqrt{||\boldsymbol{p} - \boldsymbol{p}'(\xi)||^2} \tag{3.14}$$

of the projections $\boldsymbol{p}$ used for reconstruction and the estimated projections $\boldsymbol{p}'(\xi) = \boldsymbol{W}\boldsymbol{x}_v(\boldsymbol{\xi})$ serves as the scalar function to be optimized.

Using the arbitrarily set value $\delta = 0.2$ for the finite differences, the gradient direction is estimated according to (2.27), the current parameter value is varied by adding and subtracting δ for each parameter value in the parameter vector and evaluating (3.14) for the new values.

To only consider estimates ξ_{est} that improve the estimates, the contribution to the approximated gradient is set to 0 if neither new parameter value yields a lower or equal projection error.

The approximated gradient is thus

$$\nabla = \mathrm{RMSE}(\xi_i + \delta) - \mathrm{RMSE}(\xi_i - \delta) \tag{3.15}$$

The new estimate ξ_{new} is computed using (2.28) with the approximated gradient and a step size of 1.

If the error is lower or equal to the error of the previous estimate, ξ_{new} is the new parameter vector of the fiber. If it is not lower, the delta value is decreased to 75% of its current value. This process is repeated for a minimum of $nmin = 18$ times and a maximum of $nmax = 35$ times. The iteration stops

either at the upper limit of repetitions or if the rate of change of the error, defined as

$$\rho_i = 1 - \frac{\mathrm{RMSE}(\xi_i)}{\mathrm{RMSE}(\xi_{i+1})}, \tag{3.16}$$

is lower than 0.001.

3.5 Simulations and Experiments

To test the method described above, two different fiber samples were simulated. In the following, the process of generating the fiber models and rendering the models into a voxel grid is described. Subsequently, the experimental setup is explained and finally the results of those experiments are shown.

3.5.1 Simulation of GFRP Phantoms

Glass fiber-reinfored materials are made up of some matrix material, usually a polymer, and the glass fibers themselves. Those glass fibers can be unordered, as well as highly aligned. To simulate a sample of such a material, the fibers need to be placed in some space in a way they do not intersect. To that end the random sequential adsorption (RSA) [91] algorithm is employed here. A pseudo-code version of the algorithm is shown in Algorithm 3.

The algorithm "adsorbs" shapes by randomly generating them and then testing if they intersect the already placed shapes. If there is no collision with any other shape, the shape is appended to the list of generated shapes. Otherwise the shape is moved randomly without changing the other parameters and the process is repeated at most *num_tries* times. Shapes are placed until a pre-defined number of shapes is reached. In cases where no more shapes can be placed, but where the desired number of shapes has not yet been reached, the user can provide a maximum number of iterations. The maximal amount of shapes that can be adsorbed is called the "saturation limit" and varies based on the shape being adsorbed [92].

More complex approaches exist, like the mechanical migration method introduced by Schneider [93] or the random-walk based method by Altendorf & Jeulin [94], which can increase the packing density over RSA. These approaches were omitted in this work, as the simulation provided sufficiently accurate fiber volumes and real data by nature has representative packing density.

Algorithm 3 Random Sequential Adsorption

```
1: procedure RSA(num_tries, num_shapes)
2:     shapes ← empty_list()
3:     while length(shapes) < num_shapes do
4:         new_shape ← new_random_shape()
5:         for i = 0; i < num_tries; i++ do
6:             if not collides(s, new_shape) ∀s ∈ shapes then
7:                 shapes.append(new_shape)
8:                 break
9:             else
10:                new_shape.move_random()
11:            end if
12:        end for
13:    end while
14:    return shapes
15: end procedure
```

3.5.2 Testing Collision of Fibers

For the RSA algorithm to properly work for fibers, it needs to be possible to find out if two fibers collide. A collision of two cylinders, which is the shape used to represent the fibers, is expensive to compute exactly. Computing if two lines intersect is not hard [95]. Adding a radius to the lines changes the computation from requiring an intersection point (or whether the lines intersect) to finding the smallest distance between the lines and finding if their distance is smaller than their combined radii. Making those tubes cylinders, adds additional logic to find if they intersect in the body of the cylinder or if the closing faces on either end do [96]. Especially when thousands of collision tests need to be performed, as is the case in RSA, some simplifications need to be made to keep the computation time low.

Using an inexact computation simplifies this problem a lot. If the cylinder is instead seen as a "capsule", i.e. a cylinder with half domes on either end instead of flat circular ends, the collision equation reduces to computing the distance between the central line segments of the two cylinders. Let the two line segments be expressed by two points in space

$$L_1 = (1-s)\boldsymbol{p}_0 + s\boldsymbol{p}_1$$
$$L_2 = (1-t)\boldsymbol{q}_0 + t\boldsymbol{q}_1,$$

with the loci $s \in [0,1]$ and $t \in [0,1]$. The radii of the fibers are r_{L_1} and r_{L_2}, respectively. The minimal distance between those lines is given by the connection of two points: one on each of the lines, such that the connecting line is perpendicular to both lines. The (squared) length of this line segment is expressed as a function of s and t

$$R(s,t) = |L_1(s) - L_2(t)|^2 = as^2 + 2bst + ct^2 + 2ds - 2et + f, \tag{3.17}$$

with

$$\begin{aligned} a &= (\boldsymbol{p}_1 - \boldsymbol{p}_0)\cdot(\boldsymbol{p}_1 - \boldsymbol{p}_0), \quad b = (\boldsymbol{p}_1 - \boldsymbol{p}_0)\cdot(\boldsymbol{q}_1 - \boldsymbol{q}_0) \\ c &= (\boldsymbol{q}_1 - \boldsymbol{q}_0)\cdot(\boldsymbol{q}_1 - \boldsymbol{q}_0), \quad d = (\boldsymbol{p}_1 - \boldsymbol{p}_0)\cdot(\boldsymbol{p}_0 - \boldsymbol{q}_0) \\ e &= (\boldsymbol{q}_1 - \boldsymbol{q}_0)\cdot(\boldsymbol{p}_0 - \boldsymbol{q}_0), \quad f = (\boldsymbol{p}_0 - \boldsymbol{q}_0)\cdot(\boldsymbol{p}_0 - \boldsymbol{q}_0) \end{aligned}$$

The two fibers intersect if the distance $R(s,t) \leq r_{L_1} + r_{L_2}$, so minimizing $R(s,t)$ will yield the desired solution. Equation 3.17 can be rewritten as the quadratic function

$$R(s,t) = \boldsymbol{s}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{s} + 2\boldsymbol{K}^{\mathrm{T}}\boldsymbol{s} + f, \tag{3.18}$$

with

$$\begin{aligned} \boldsymbol{K} &= \begin{bmatrix} d \\ -e \end{bmatrix} \\ \boldsymbol{M} &= \begin{bmatrix} a & -b \\ -b & c \end{bmatrix} \\ \boldsymbol{s} &= \begin{bmatrix} s \\ t \end{bmatrix}. \end{aligned}$$

The minimum of R is at a point where the gradient is 0

$$\nabla R = 2\boldsymbol{M}\boldsymbol{s} + 2\boldsymbol{K} = 0. \tag{3.19}$$

The loci at which the minimum is reached are computed as

$$\boldsymbol{s} = -\boldsymbol{M}^{-1}\boldsymbol{K} = \frac{1}{ac - b^2}\begin{bmatrix} be - cd \\ ae - bd \end{bmatrix}. \tag{3.20}$$

The numerical implementation of this solution is described in detail in

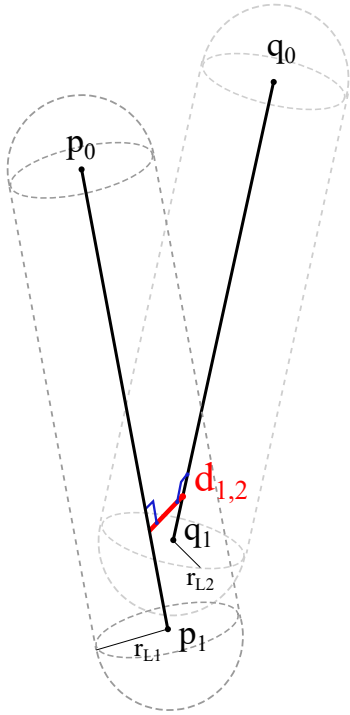


FIGURE 3.7: Schematic of two capsules and the geometrical features needed to compute if they intersect or not. Shown in dashed lines are the contours of the cylinders and their half dome end caps. The thick black line represents the center line of the cylinder. Shown in red is the shortest connection between those center lines.

[97]. Figure 3.7 shows the important geometrical features needed for computing the intersection of the capsules.

3.5.3 Voxelization of Fiber Models on a Cubic Lattice

The above description of the optimization of fiber parameters has the prerequisite for a function to exist, that produces projections from the fiber models. In this context such a function can exist, but it is hard to find. The solution employed here is the rendering of the fiber model onto a voxel lattice grid that corresponds to the lattice grid of the reconstruction, such that the projections of this rendered lattice matches up with the projections that produced this reconstruction. This of course requires the geometry to match as well.

It is assumed that the geometry is known completely, as it is also a requirement for a successful algebraic reconstruction (see Subsection 2.4.2). The fiber model is translated to voxels by testing per voxel, whether or not this particular voxel is located inside the cylinder c describing the fiber

$$c(x,y,z) = \begin{cases} \mu_{\text{fiber}} & \text{if } y^2 + z^2 \leq r_{\text{fiber}}^2 \text{ and } |x| \leq \frac{l_{\text{fiber}}}{2} \\ \mu_{\text{polymer}} & \text{otherwise} \end{cases}. \tag{3.21}$$

The fiber cylinder is first transformed in such a way that the x-axis aligns with the main axis of the cylinder. Then the above function is sampled at integer voxel positions within the voxel grid and the pixel values set to the corresponding result, either μ_{fiber} or μ_{polymer}. To achieve smooth transitions between fiber material and polymer material, the voxels located inside the fiber are sub-sampled n times, resulting in n^3 samples per voxel. The corresponding voxel is set to the average value of the subsamples obtained that way. If a voxel is already set to a non-background value, it is not set to μ_{polymer}, but to the maximum of the existing and the newly computed value, to avoid partially erasing already placed fibers.

3.5.4 Experiments

The shown PARE algorithm was verified by extracting the fibers from several simulated volumes. To that end, the previously described voxelization procedure was used to render fibers with varying parameters into a volume of fixed size. The straight fibers were assigned random orientations drawn from a Van Mises - Fisher distribution (VMF) distribution

$$f_{p,\text{VMF}}(\boldsymbol{x}; \boldsymbol{\mu}_{\text{VMF}}, \kappa) = C_F \exp(\boldsymbol{\mu}_{\text{VMF}}{}^T \boldsymbol{x}), \tag{3.22}$$

where

$$C_F = \kappa / (4\pi \sinh \kappa), \tag{3.23}$$

where the vector $\boldsymbol{\mu}_{\text{VMF}} = (\alpha, \beta)$ denotes the mean direction of the distribution and κ the *concentration parameter*, where a large value of κ corresponds to a lower variance, i.e. a higher concentration around the mean direction [98]. This is also illustrated in Figure 3.8. The fiber center positions are also randomized in a box smaller than the target volume, so that the fibers do not protrude out of the volume boundaries.

Two distinct phantoms with 100^3 voxels were generated. The target for the RSA algorithm was 150 fibers. Due to the random tries for placing the

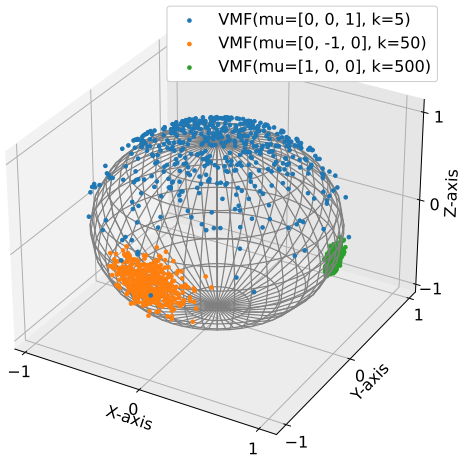
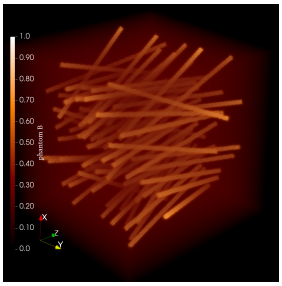


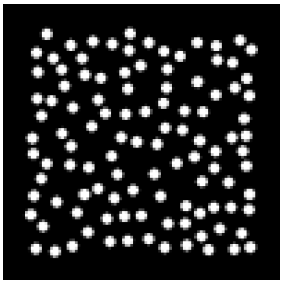
FIGURE 3.8: Illustration of three expressions of VMF distributions shown on the unit sphere as point samples.

fibers, not all of the fibers will be placed. Depending on the parameters for the fiber distribution, the adsorbed volume will ultimately contain a varying number of fibers. After processing, phantom A contained 109 fibers drawn as described from a VMF distribution with $\boldsymbol{\mu}_{\text{VMF}} = (\frac{\pi}{2}, 0)$ and $\kappa = 40$. The fibers for phantom B were drawn from a VMF distribution with $\boldsymbol{\mu}_{\text{VMF}} = (0, 0)$ and $\kappa = 7$. It contained only 72 fibers after RSA was finished because of the greater variance in the distribution. The intensities for both the fibers and the background polymer matrix were manually estimated from an existing real tomogram of a GFRP sample. The fiber attenuation was estimated to be $\mu_{\text{fibers}} = 0.76 \pm 0.05$ and the polymer attenuation to be $\mu_{\text{polymer}} = 0.23 \pm 0.07$. The mean values of those estimated attenuation values was taken for the simulated phantoms prior to adding noise. Phantom B is shown in Figure 3.9A together with a central slice of phantom A (Figure 3.9B) and the same central slice of a reconstruction of phantom A (Figure 3.9C).

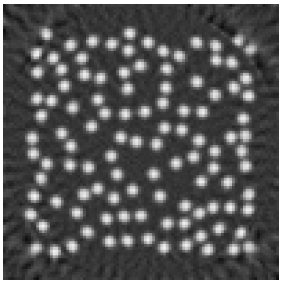
From those phantoms, forward projection images were created using a simulated cone-beam geometry in the ASTRA toolbox, as that is the most commonly used geometry in industrial and desktop X-ray scanners. The



(A) Ground truth of the synthetic phantom B with 72 individual fibers.



(B) Central slice of phantom A with 109 individual fibers of varying position, direction and length.



(C) Central slice of the reconstruction of the phantom shown in (b) from 100 simulated projections.

FIGURE 3.9: Full rendering and the central slices of the generated phantoms and the reconstruction after adding noise. Figure reproduced from [77] under CC 4.0.

phantom was placed in the origin of the system. The source-detector distance (SDD) was 250 mm and the source object distance (SOD) was 14 mm. The simulated detector had square pixels with a size of 50 µm. This yielded an effective (isotropic) detector pixel size of 2.8 µm in the reconstructions, with a magnification of around 17.86 in the center plane of the phantom. In Figure 3.9b and Figure 3.9c the central slice along the yz-plane of phantom A and the same slice of a reconstruction of said phantom from simulated projections are shown.

Using the generated data, the performance of the PARE method was evaluated as a function of both the number of projection angles available and the SNR. In all cases 100 SIRT iterations were used as the base line for the reconstructions of the two generated phantoms. The first experiment was performed with additive, Gaussian distributed white noise, which was added to the projection data before the reconstruction. The detection and subsequent optimization was repeated for each noise level σ, as indicated in Table 3.1.

3.5.5 Finding fiber pairs between two sets

The simulation experiments enable the comparison of the fiber estimation quality compared to the ground truth of the fibers in the phantom. To be able to compare the fibers properly, a mapping between a set of detected and/or optimized fibers and the ground truth set needs to be found. Regarding each

σ	SNR_A	SNR_B
0.5	17.06 dB	16.88 dB
1.0	14.05 dB	13.87 dB
2.0	11.04 dB	10.86 dB
3.0	9.27 dB	9.10 dB
4.0	8.03 dB	7.85 dB
5.0	7.06 dB	6.88 dB

TABLE 3.1: SNR for the noise levels used in our experiments for both phantoms. Table reproduced from [77] under CC 4.0.

fiber in either set as a vector in a parameter space, the fiber parameter vectors corresponding to each other between the two sets are the ones closest to each other in Euclidean space. Mathematically, the one-to-one mapping performed can be described as follows. Let the sets of fiber parameter vectors be F_{gt} and F_{est} the ground truth and estimated parameters, respectively. The set F_{gt} is the set of all reference fiber parameters a_n. The mapping from one set to the other is iteratively defined as

$$f_{a_n} = \arg \min_{b \in F_{\text{est}} \setminus \{f_{a_1}, \ldots, f_{a_{n-1}}\}} ||a_n - b||^2. \tag{3.24}$$

The first link

$$f_{a_1} = \arg \min_{b \in F_{\text{est}}} ||a_1 - b||^2. \tag{3.25}$$

This implies that the mapping depends on the order of processing if two or more fiber parameter vectors from one set have the same distance to one single fiber parameter vector in the other set. This case is assumed to be unlikely, and even if it occurs, the error value will presumably be the same for all of them, so the order is not important. If there were less or more fibers detected than are in the ground truth, only the parameter vectors that fit best are mapped and the rest is discard as not detected. The comparison is only done for the parameters of detected fibers that have a partner in the ground truth set.

3.5.6 Performance in presence of noise

In Figures 3.11, 3.12 and 3.10 the length, direction and position errors are shown in function of the standard deviation σ of additive noise we added to

the projection data. As expected, the errors increase with increasing σ. The length estimates are barely changing for the lower noise levels $\sigma = 0.5$ and $\sigma = 1.0$ and are in the same range as the errors for 100 projections in the previous tests. This is also expected, as we used 100 projections consistently for this experiment. The SNR for the different noise levels and phantoms are laid out in table 3.1. The SNR value is computed with the following equation

$$\mathrm{SNR} = 10 \log \left(\frac{\mu_{\mathrm{signal}}}{\sigma_{\mathrm{noise}}} \right), \tag{3.26}$$

where μ_{signal} is the mean of the measured intensity of the projections and σ_{noise} the corresponding noise level.

The length and the centroid estimates seem to be more affected by the noisy projections than the direction estimates. In the case of the highest noise level, the length estimate is 2 voxels too large in the upper quartile for phantom A and around 1 voxel for phantom B.

This is most likely due to the way voxels change in the simulated projections of the model. When varying the direction vector, more voxels change their value compared to when the length or centroid position is changed. This in turn means that the optimization is more sensitive to small changes in direction, especially when the fibers are very long.

3.5.7 Performance varying the number of projections

In Figures 3.13,3.14 and 3.15 the quality of the estimation with PARE in function of the number of projections is shown. In all figures there are two box plots for each projection, where the black boxplot refers to results for phantom A and the orange one corresponds to phantom B. It can clearly be seen that the algorithm can retrieve the individual fiber centroids with around ± 0.5 voxel accuracy in the upper and lower quartiles, even with as low as 30 projections for both phantoms.

It can be observed in Fig. 3.15 that the errors in the coordinate direction that corresponds to the mean axis of the direction distribution are higher. While the direction estimation is not affected by this, the length estimation and centroid estimation are correlated. The length estimation can retrieve the fiber length up to ± 1 voxels for 30 projections. The direction vector can be approximated to about $0.6\,^{\circ}$ for the upper quartile.

This error naturally decreases with an increasing number of projections, as there is more information available for computing the projection error making the procedure more sensitive to small parameter changes.

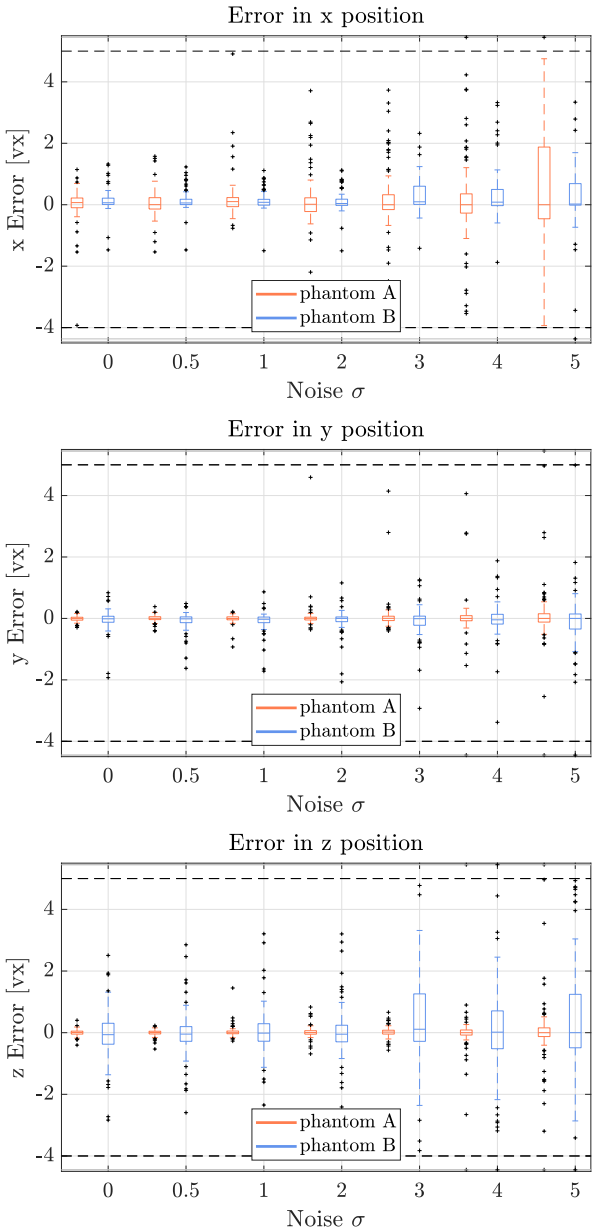


FIGURE 3.10: Centroid position error for for several noise levels on both phantoms A and B on the estimated centroid position of the fiber. Outliers were capped at ±4 voxels, but are still shown outside the horizontal dotted lines. Figure reproduced from [77] under CC 4.0.

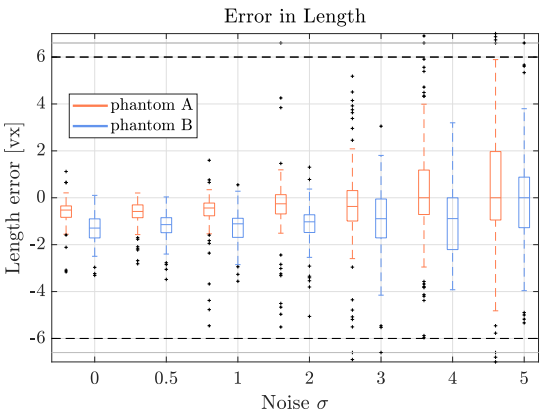


FIGURE 3.11: Length error for several noise levels on phantoms A and B with respect to the estimated fiber length. Outliers were capped at ±6 voxels, but are still shown outside the horizontal dotted lines. Figure reproduced from [77] under CC 4.0.

With 100 projections, the error for the centroid position is as low as ±0.3 voxels which is around the accuracy of the sub-sampling we do for the voxelization of the fibers in the phantoms. The direction can be estimated to around 0.4 ° for phantom A and 0.25 ° for phantom B. Lengths are estimated between 0.2 and 0.7 voxels for phantom A and between 0.9 and 1.8 voxels for phantom B.

3.6 Discussion

The PARE algorithm demonstrates the use of a parametric fiber model. The parameters of that model are estimated directly using the available projection domain, thereby largely avoiding reconstruction artifacts that may otherwise influence the fiber position and direction estimation. As a result, the parameter estimation is robust even for a very small number of projections. Most algorithms trying to estimate fiber parameters use several thousands of projection images to compute quantities on their fiber specimen [68, 71, 73].

However, PARE is limited by a couple of factors. The rigid cylinder model is adequate for fibers that are not bent, which is a reasonable assumption in GFRPs that have moderate aspect ratios. In case of high aspect ratio fibers, the model would need to be extended to allow bending. Altendorf and Jeulin

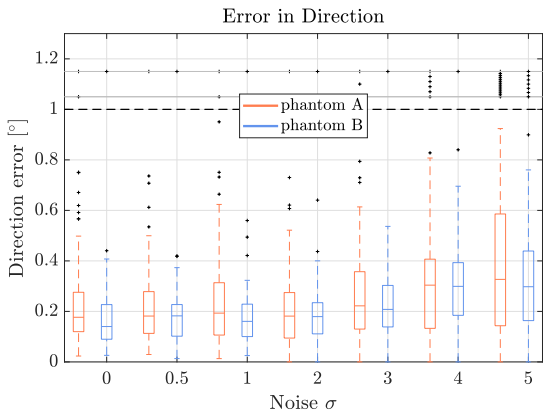


FIGURE 3.12: Direction error for for several noise levels on phantoms A and B with respect to the estimated direction vector. Outliers were capped at 1 °, but are still shown outside the horizontal dotted lines. Figure reproduced from [77] under CC 4.0.

proposed an approach to model fibers as short fiber segments on a chain and generated random fiber networks from it using a random walk approach [94]. A similar model of cylinders chained together for example, could be used to represent the fibers in our approach, but would require heavy modification of the Hough transform or a different approach for the clustering of detected fiber center lines altogether. As the Hough transform can be defined for an arbitrary parameterized curve [79], the model could be transformed to approximate weaving in carbon fibers for example. This would of course increase the number of parameters in the Hough accumulator exponentially and therefore might not be practical for very complicated fiber systems. The sampling of the fiber direction already poses problems in the case of very short fibers, which can be detected with a high deviation from their true direction. In the following chapters further techniques will be introduced that address some of these shortcomings and extend the model to curved fibers. The optimization algorithm is also going to change to accommodate the new model.

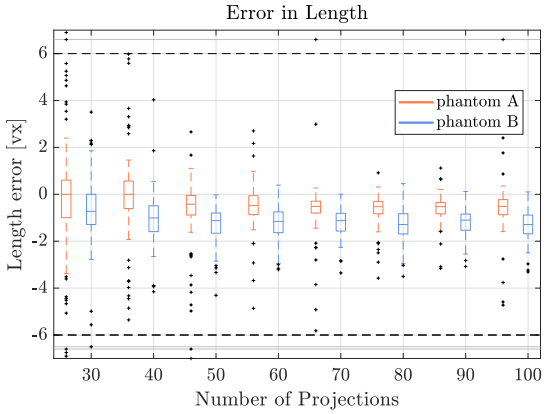


FIGURE 3.13: Length error for varying number of projections on phantoms A and B with respect to the estimated fiber length.Outliers were capped at ± 6 voxels, but are still shown outside the horizontal dotted lines. Figure reproduced from [77] under CC 4.0.

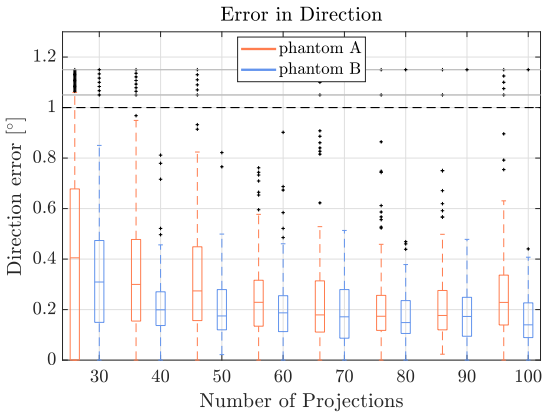


FIGURE 3.14: Direction error for varying number of projections on phantoms A and B with respect to the estimated direction vector.Outliers were capped at 2 °, but are still shown outside the horizontal dotted lines. Figure reproduced from [77] under CC 4.0.

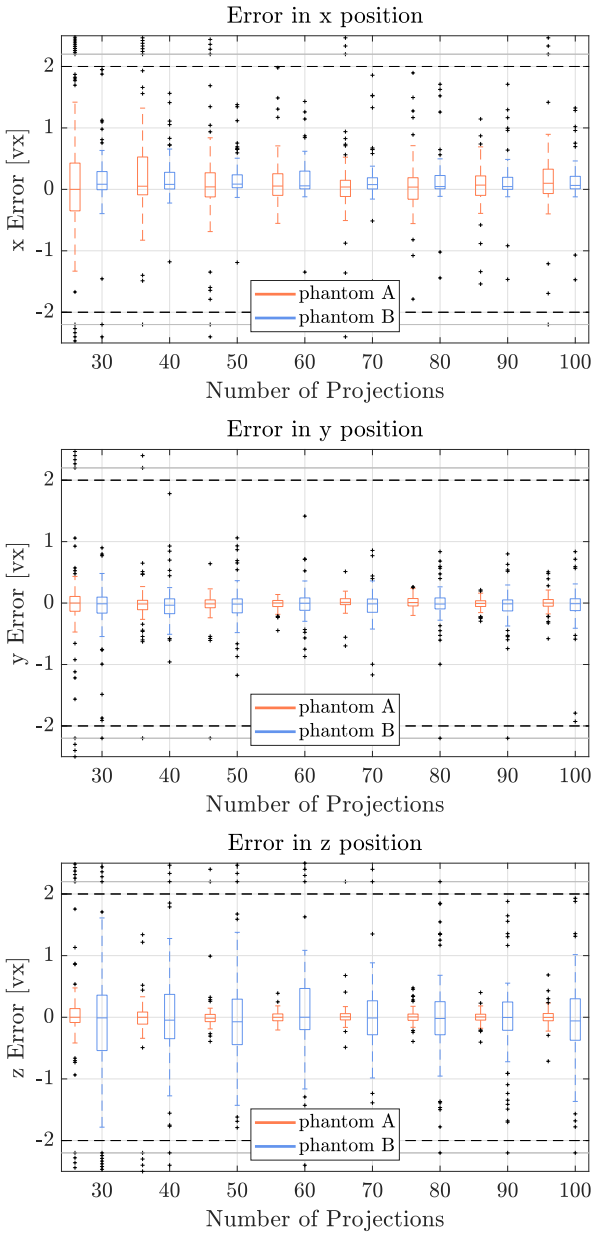


FIGURE 3.15: Centroid position error for varying number of projections on both phantoms A and B on the estimated centroid position of the fiber. Outliers were capped at ±2 voxels, but are still shown outside the horizontal dotted lines. Figure reproduced from [77] under CC 4.0.

Chapter 4

Parametric Reconstruction of Curved Fibers

The PARE method presented in the previous chapter shows clearly that it is possible to increase the fiber detection accuracy of single fiber detection methods by optimizing the fiber model based on the projection data. The method served as a proof of concept, but was not applied to any real data. The main reason here is that the model does not accommodate the morphology of real fibers.

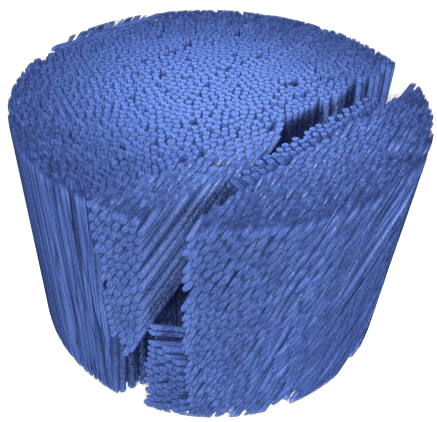


FIGURE 4.1: Example of a GFRP tomograph showing straight fibers in various orientations. The dataset was acquired using a ZEISS Xradia 520 Versa [99] and shows a sample from a glass fiber-reinforced crimp fabric. These materials help increasing strength in wind turbine blades.

Even straight fibers, like the one shown in Figure 4.1 have small deviations from the perfect straightness. For the straight fibers, optimization should be possible. However, the more the fibers become curved, the more this curvature needs to be modeled to enable optimization. In the following sections the work towards a new framework curved Parametric Reconstruction (cuPARE) is laid out.

4.1 Introduction

Many GFRP analysis methods focus on a specific feature of the fiber-reinforced composite, such as the orientation or length distribution of the fibers. Zauner et al. [73] use a template matching and binary thinning to extract fiber center lines and modeling the fibers in composites from them. On fiber bundle level, Bhattacharya et al. [100] introduced MetaTracts, a tractography approach based on Hessian analysis and hierarchical clustering. Emerson et al. [101] apply semi-automatic dictionary learning (InsegtFibre) for the segmentation of fiber voxels from uni-directional GFRP samples and detect their centers per slice, stacking the resulting points on top of each other, subsequently clustering into individual fiber traces. While tracking individual fibers requires higher resolution than methods that extract parameter distributions, it provides the opportunity to extract various statistical measures without having to re-analyse the tomograms.

A compact parametric description of the fibers contained in a GFRP sample is therefore convenient. Indeed, parametric models are easier to manipulate than a list of coordinates describing the center line and take significantly less memory to store than the voxel grids. Several ways of describing fibers as parametric curves in 3D space have been proposed. Adluru et al. [102] use cosine series to represent and match long fiber traces from magnetic resonance imaging tractography data and Lemkaddem et al. [103] employ splines for a similar purpose. Zhao et al. [104] represent single fibers in yarn plies with circular helices. The stress analysis following the extraction of the fibers is often carried out using either finite element analysis [23] as a general purpose approach or mathematical models tailored to the application [105], for which a parametric representation could be used.

Parametric Reconstruction (PARE) is a theoretical framework for estimating such parametric representations of individual straight fibers from tomographic reconstructions of GFRP samples and optimizing their orientation, location and length based on the projections used for the reconstruction [77].

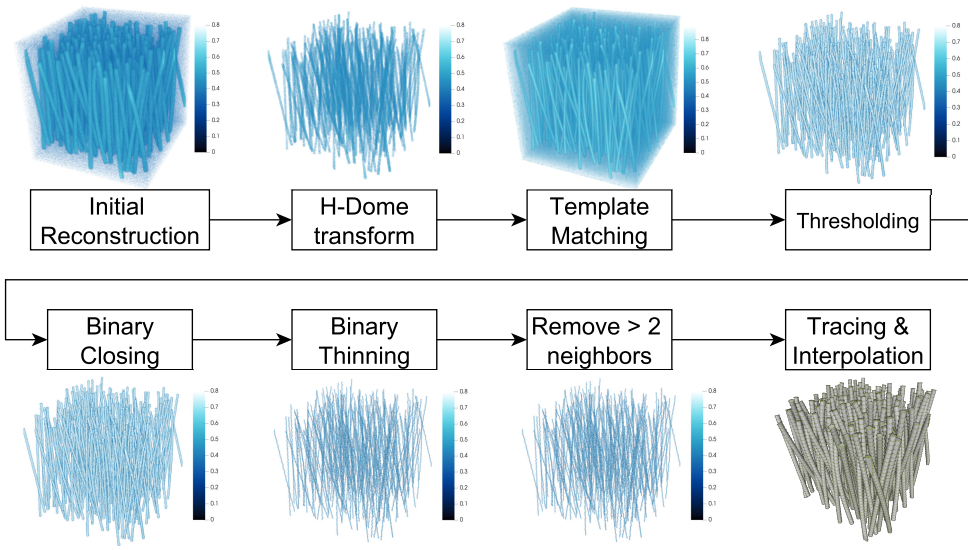


FIGURE 4.2: Image processing steps to convert a GFRP reconstruction to a fiber model. Figure reproduced from [106] under CC 4.0.

It offers a convenient way to improve upon fiber traces by optimizing directly in projection space. That is, PARE allows to retrieve fiber parameters even if only few projections are available. The validity of the PARE framework has been shown on simulated data. Unfortunately, it has only been developed for perfectly straight fibers, and has not yet been demonstrated on real data.

In this work, a new framework called curved Parametric Reconstruction (cuPARE) is proposed. cuPARE extends the fiber model to *curved* fibers and estimates the parameters of that model within an automatic processing pipeline from reconstructions from only a few projections. The individual steps of the framework are visualized in Figure 4.2 and explained in more detail in the following sections.

4.2 Curved Fiber Model

A first step towards making curved fibers detectable and optimizable, is to extend the fiber model to include more parameters. In PARE the fiber is represented by a centroid in cartesian coordinates, the direction vector relative to that centroid in spherical coordinates, and a length. With these 6 scalars it

is possible to describe any straight fiber. However, adding curvature to this model is not straight forward. One solution might be to add an offset vector perpendicular to the fiber orientation vector and displace the centroid, but not to displace the ends. This would add 2 more scalars, if using spherical coordinates. But what if the fiber forms an S-curve? The addition of further deviations in the fiber shape will make this style of model more and more complicated and less and less flexible.

The simplest way is to take the data as is and represent a single fiber by a radius and a collection of point coordinates that make up the center line. This model is very flexible as each individual point can be moved. However, the flexibility comes at the cost of cumbersome manipulation.

4.2.1 Arc-length Parameterization

A more extensible solution is the introduction of a parametric curve, represented by a function of the form $f : \mathbb{R} \rightarrow \mathbb{R}^3$, mapping some scalar indicating the position along the curve to a coordinate in 3D space. The modeling would then require finding the parameters of an appropriate function f to represent a single fiber. This kind of representation is called ”arc-length parameterization” because the scalar parameter of the function describes the position or *locus* along some curve. Normalizing the parameter by the length of the curve leads to the scalar always lying in the interval [0, 1], which can have advantageous effects in the optimization that will follow later. A single step size along the gradient direction can provide different sized steps in parameter values, which can lead to a higher number of iterations being required to get the optimization to converge [42].

Of course, there already exist several different representations for curves in space, two of which subsequently will be explained and investigated with regards to their suitability to the presented problem. Approaches not using arc-length parameterization, like splines, which fit functions piece-wise between the given samples, also have been used. Splines will not be considered in this thesis, as they usually require more parameters and are not as easily manipulated, which will be necessary for the optimization procedure.

4.2.2 Cosine Series Representation

Cosine Series Representation (CSR) is a arc-length parameterization method that uses coefficients of a cosine series expansion to represent curves. They

were introduced by Adluru et al.[102] to represent white matter fiber bundles for Diffusion Tensor Imaging [107], a technique which is part of Magnetic Resonance Imaging (MRI). A more detailed overview of this technique follows in Subsection 4.4.1.

Given a set of coordinates $\boldsymbol{\zeta} = (\zeta_x, \zeta_y, \zeta_z)$ that lie on some curve to be found, this method estimates a least square fit to those points using a cosine series via Fourier coefficients in a search space

$$\mathcal{H}_k = \left\{ \sum_{l=0}^{k} c_l \psi_l(t) : c_l \in \mathbb{R} \right\} \subset \mathcal{L}^2[0,1], \tag{4.1}$$

which spans up to the k-th degree eigenfunctions $\psi_0, \psi_1, \ldots, \psi_k$. The eigenfunctions are Fourier cosine basis functions

$$\psi_l(t) = \sqrt{2}\cos(l\pi t). \tag{4.2}$$

The k first functions (ordered by their eigenvalues $\lambda_l = l^2\pi^2$) form the orthonormal basis of the entire space of square integrable functions $\mathcal{L}^2[0,1]$. This way the least squares solution for a curve fitting the given points only needs to consider the most important basis functions of that subspace. Thus, the solution for the estimated curve is given by

$$\hat{g} = \arg\min_{f \in \mathcal{H}_k} ||f - \zeta||^2, \tag{4.3}$$

where the norm is defined as

$$||\psi|| = \langle \psi, \psi \rangle^{1/2}.$$

This solution can be represented as the k-th order series expansion

$$\hat{g} = \sum_{j=0}^{k} \langle \boldsymbol{\zeta}, \psi_j \rangle \psi_j. \tag{4.4}$$

This minimizes the distance between all points along the curve, represented by the locus u, and some function f. The coefficients of the representation are the weights for each of the basis functions, which together form this function

f. As a function of the normalized arc-length the curve becomes

$$g(u) = \sum_{j=0}^{k} \begin{pmatrix} a_{x_j} \\ a_{y_j} \\ a_{z_j} \end{pmatrix} \psi_j(u), \tag{4.5}$$

where the coefficients $a_{(x,y,z)_j}$ can be directly estimated in a matrix expression. The matrix has the size $k \times 3$

This representation is able to adequately model the morphology of very complex traces of white matter in the brain and can easily be scaled in the degrees of freedom by changing the maximum degree of the eigenfunctions.

4.2.3 Polynomial Representation

Similarly, polynomials can be used for the representation of curves. By using one polynomial of degree k per coordinate direction g_x, g_y, g_z that described the projection of the curve onto the xy, zy and xz plane. Exactly like with the CSR, they form a vector equation that yields the coordinates ζ of the curve as a function of the (normalized) location u.

$$g(u) = \begin{pmatrix} g_x(u) \\ g_y(u) \\ g_z(u) \end{pmatrix} = \begin{pmatrix} \sum_{j=0}^{k} a_{x_j} u^j \\ \sum_{j=0}^{k} a_{y_j} u^j \\ \sum_{j=0}^{k} a_{z_j} u^j \end{pmatrix}. \tag{4.6}$$

The parameters of the polynomial can be estimated in a least squares sense from a set of given coordinates by minimizing the error just like in Equation 4.3, where the function f is a polynomial of degree k. However, the higher the degree of the polynomial, the higher the likelihood of the curve fit being inadequate. This is caused by a phenomenon called "Runge's phenomenon", which describes the behavior of the polynomial at the ends of the provided curve points, given that they are equidistant. The higher degree polynomials tend to deviate very quickly from the estimated curve outside of the domain of the provided points [108]. To test which of the models is best suited to be used as fiber center line representation, the quality of fit was investigated.

4.2.4 Parameter estimation with polynomials and CSR

To investigate the quality of fit, a dataset of estimated fiber center lines was used. The center lines were detected in the dataset shown in Figure 4.1 using

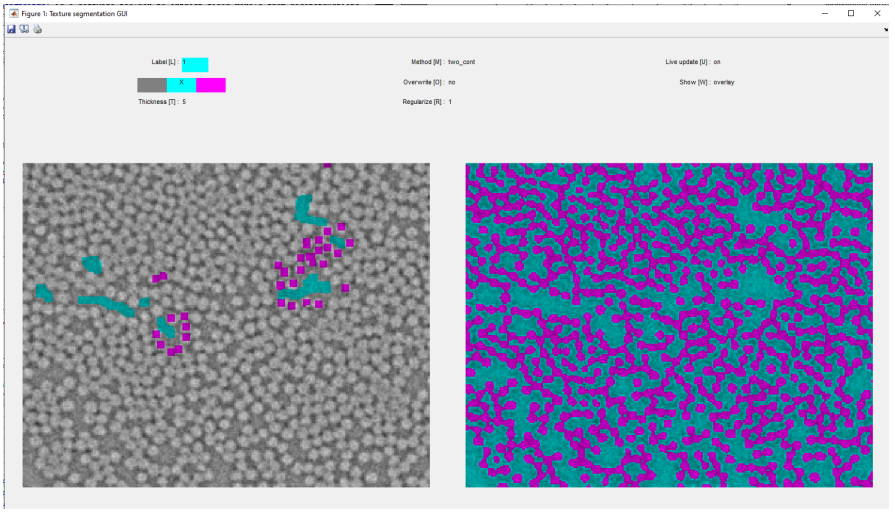


FIGURE 4.3: Segmentation Graphical User Interface (GUI) of InsegtFibre with an example region loaded and partially annotated. Marked in cyan, the background is shown, the foreground is annotated in pink.

the InsegtFibre framework [101].

InsegtFibre is a software toolbox to extract fiber voxels from reconstructions using a supervised dictionary learning method and provides functionality to track individual fibers through time sequences of reconstructions. The fiber voxel extraction takes a user input in form of in-painting of regions of the image representing the background and foreground, i.e. the fibers. The input informs the dictionary learning procedure and the tool shows a suggestion of segmented fiber voxels. The more in-painted regions are provided, the more accurate the segmentation of the fiber voxels. Figure 4.3 shows the interface of InsegtFibre with some fibers segmented out by hand on the left, and a proposed segmentation on the right. The dataset used is the same as in Figure 4.1.

Both types of representation were then fitted to the fiber traces obtained with InsegtFibre, shown in Figure 4.4, with varying degrees of freedom k. The quality of fit was then evaluated by calculating the average distance measure

$$d_{a,b,min} = \frac{1}{m}\sum_{j=1}^{m} \mathrm{dist}_{\min}(\zeta_{a,j}, \zeta_b), \tag{4.7}$$

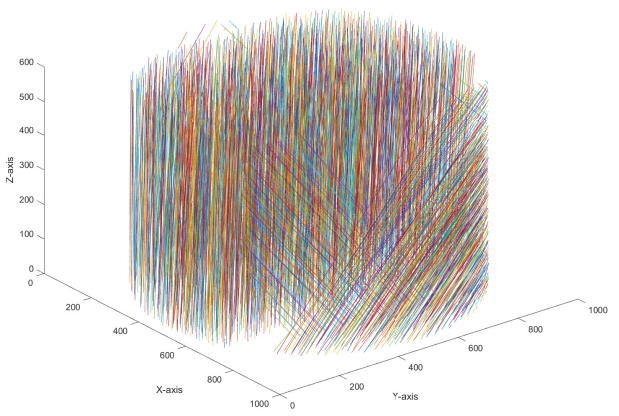


FIGURE 4.4: Fiber traces acquired using InsegtFibre from the slices of the reconstruction shown in Figure 4.1.

where $\text{dist}_{\text{min}}(\zeta_{a,j}, \boldsymbol{\zeta}_b)$ is the minimum distance between a sample point $\zeta_{a,j}$ from m sample points and a curve $\boldsymbol{\zeta}_b$ given by line segments, connecting all available samples for the curve. This measure will be referred to as *dminavg*, subsequently. Similarly, replacing dist_{min} by dist_{max}, the maximum distance was used to estimate the maximum deviation from the curve. It is referred to as *dmaxavg* in the following. These distance measures, as well as others, were introduced in [109]. The interpolated fiber center lines were treated as fibers with a constant radius of 5 voxels and imported into the FIAKER tool to evaluate the quality of fit of the estimated curves to the reference dataset.

FIAKER [109] is a visualization tool developed for the analysis of fiber characterization algorithms. It is used in this context for evaluating quality of fit, but was also used in the remainder of the work with cuPARE for the investigation of mistakes in the representation of individual fibers. The provides feedback on a fiber population level as well as on an individual fiber basis and allows to simultaneously visualize several datasets as progressions in an algorithm, for example.

The *dminavg* and *dmaxavg* measures were calculated for each fiber in each dataset using FIAKER, with the unaltered InsegtFibre traces as a ground truth reference [110]. The measures were calculated in units of voxels.

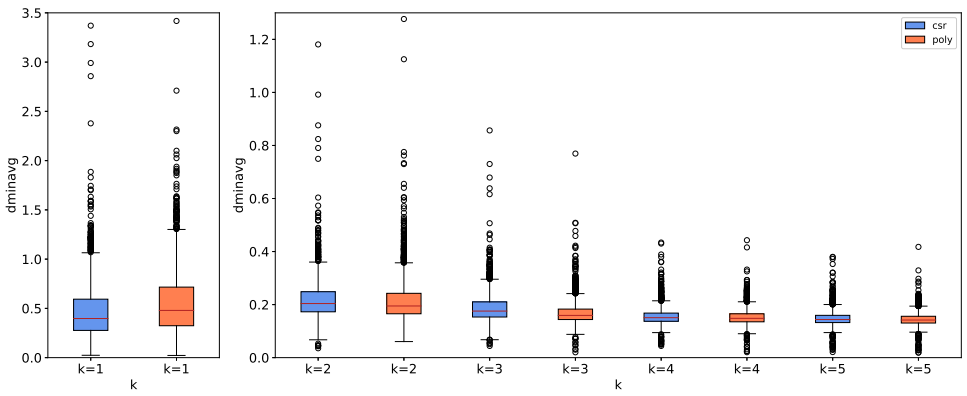


FIGURE 4.5: Minimum average distance (*dminavg*) from the ground truth of the curves estimated from the dataset shown in Figure 4.4 using polynomials and CSR as a function of the degrees of freedom k. Note that the y-axis is different for $k = 1$, to make the plots with lower errors more visible for $k > 1$.

In Figure 4.5 the minimum average distance *dminavg* between the InsegtFibre fiber traces and the CSR and polynomial interpolation is shown in function of the number of degrees of freedom k. The distances for k=1 are the highest, falling off exponentially with increasing k. The straight fibers exhibit small local deviations from the perfectly straight line, making curves with higher degrees of freedom fit better to the data. Both methods, CSR and polynomials, behave very similarly.

This changes, however, when looking at the *dmaxavg* measure in Figure 4.6. Here the polynomials fit to the data closer than the curves interpolated with CSR.

For the shown data the curvature is minimal, so the usefulness regarding interpolation of strongly curved fibers is limited. To simulate curved fibers, the given traces obtained with InsegtFibre were rotated based on their z-coordinate, such that the fibers were twisted into a helix shape. The resulting fiber traces were then interpolated as before using both polynomials and CSR. The resulting datasets are shown superimposed on each other in Figure 4.7 as visualized in FIAKER.

The same measures are shown as boxplots in Figures 4.8 and 4.9. It is clear that the added curvature significantly increases the distance of the interpolated curves from the ground truth data and that a linear model is even less suitable for those fibers, indicated by the significant difference in distance

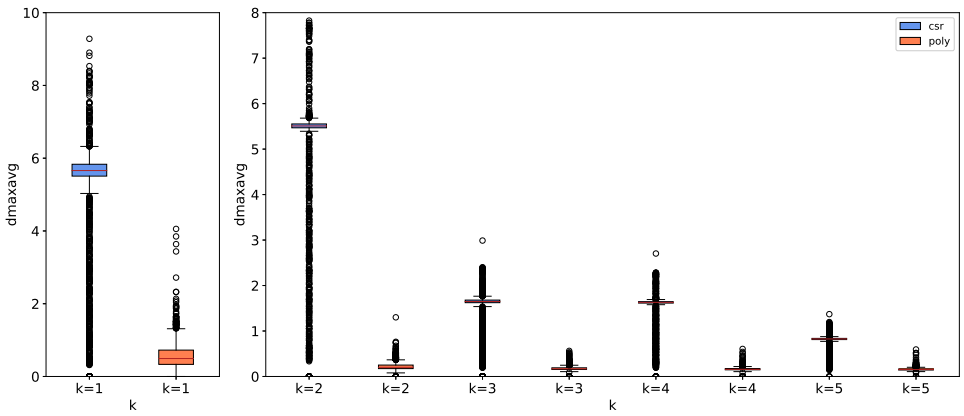


FIGURE 4.6: Maximum average distance (*dmaxavg*) from the ground truth of the curves estimated from the dataset shown in Figure 4.4 using polynomials and CSR as a function of the degrees of freedom k. Note that the y-axis is different for $k = 1$, to make the plots with lower errors more visible for $k > 1$.

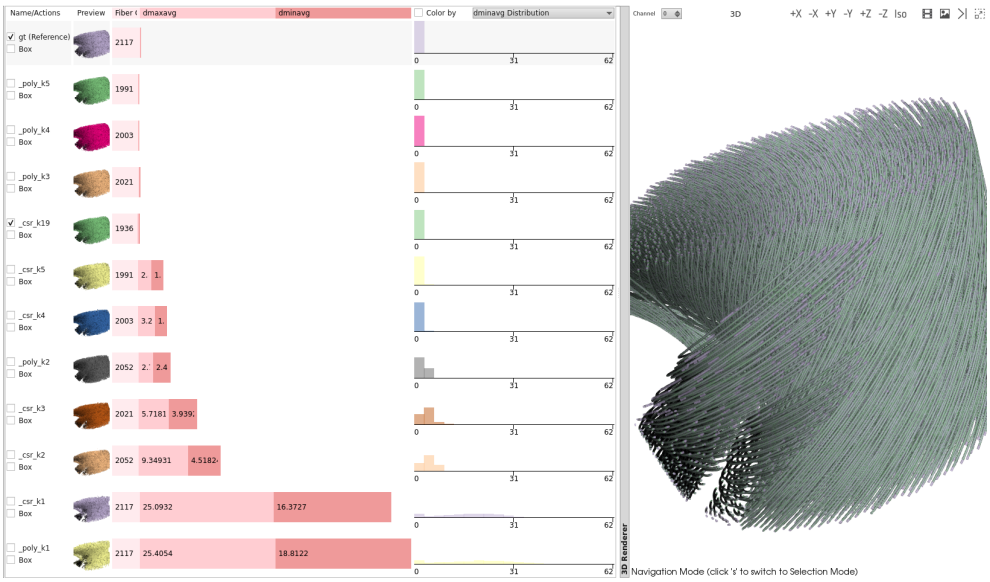


FIGURE 4.7: Screenshot of the twisted fibers produced from the straight fibers in Figure 4.4. in the FIAKER visualization and analysis tool. Shown on the left are the source dataset (gt) as well as datasets of estimated curves using CSR and polynomials with different values for k.

between the data with $k = 1$ and the rest. Like in the previous experiment, the cosine series representation has a similar lower bound as the polynomial representation, however with $k > 2$ the polynomials fit much closer to the ground truth data.

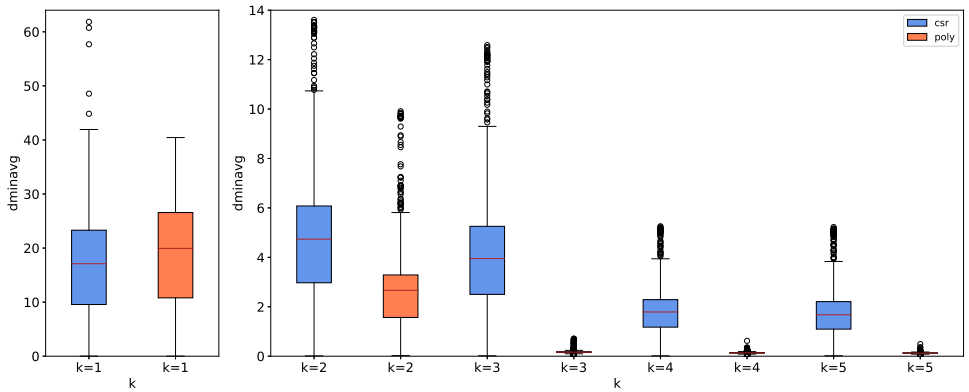


FIGURE 4.8: Minimum average distance (*dminavg*) of the curved fibers from Figure 4.7 to their interpolated counterparts as a function of the degrees of freedom k. Note that the y-axis is different for $k = 1$, to make the plots with lower errors more visible for $k > 1$.

Because of these results, the choice was made to use polynomials to represent the fibers in the model based optimization. Using the polynomial model, a fiber can be fully described by listing all coefficient values

$$\xi_j = \begin{pmatrix} a_{x_0}^{(j)}, a_{x_1}^{(j)}, \ldots, a_{x_k}^{(j)} \\ a_{y_0}^{(j)}, a_{y_1}^{(j)}, \ldots, a_{y_k}^{(j)} \\ a_{z_0}^{(j)}, a_{z_1}^{(j)}, \ldots, a_{z_k}^{(j)} \end{pmatrix} \tag{4.8}$$

and the radius.

4.3 Fiber Detection

The detection of curved fibers works similarly to the procedure described in Section 3.1. After an initial reconstruction, the non-fiber voxels are suppressed using the H-Dome transform [111] on the initial reconstruction x.

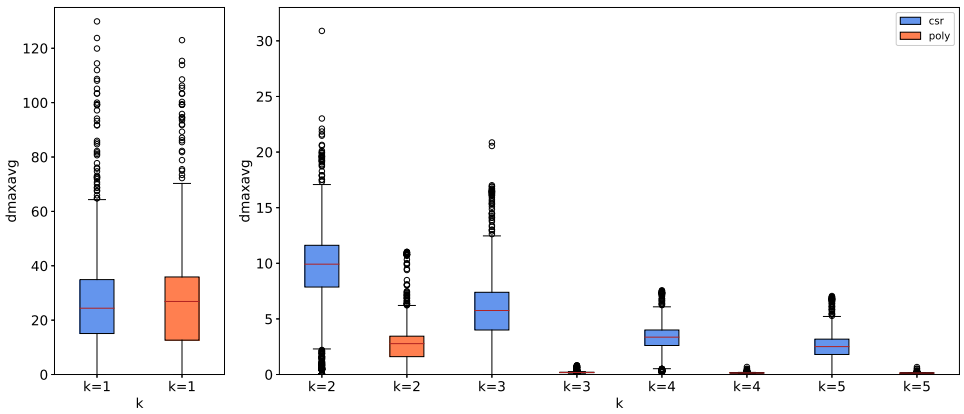


FIGURE 4.9: Maximum average distance (*dmaxavg*) of the curved fibers from Figure 4.7 to their interpolated counterparts as a function of the degrees of freedom k. Note that the y-axis is different for $k = 1$, to make the plots with lower errors more visible for $k > 1$.

The H-Dome transform is a morphological operation based on grayscale reconstruction that accentuates local maxima by removing low brightness areas lower than a predefined threshold $h \in \mathbb{R}$. Using a seed image, in this case the original reconstruction with the constant value h subtracted, a gray scale reconstruction is performed, yielding an image with local structures lower than the h suppressed. The parameter h depends on the local contrast and is chosen empirically per data set

The image with the suppressed background is then matched with a spherical template. This is a change from PARE, where the template was a Gaussian with the assumption that the reconstruction could be blurry and thus a blurry sphere would work well. In further testing it became clear that a solid sphere also works well on blurry fibers, but the reverse is not necessarily true. Therefore, a spherical template was chosen. The sphere was givena radius equal to the average fiber radius in the reconstruction. As in the previous chapter, the fiber radius is assumed to be known *a priori*. In practice, fiber radii vary slightly, but detection works well with an average value.

The local maxima in the template matching image indicate the fiber center line, which can be extracted using thresholding. The threshold is set to a value of 66% of the maximum intensity in the reconstruction by default,

but can be adapted to the reconstruction if needed. The threshold was determined empirically by computing a histogram and selecting a value close to the intensity peak, such that most of the lower intensities would be segmented out.

The fiber center lines are extracted from the resulting binary image by binary thinning. During this thinning the fiber center line can develop branches or loops, which is caused by intensity inhomogeneities in the reconstruction. To remedy this, a binary closing operation is applied to the binary image prior to the thinning operation to close any holes that might exist in the fibers. Any branching that still remains is removed by removing any non-zero voxel with 3 or more non-zero neighbors. The locations at those voxels are then traced to acquire the trajectories of the fibers.

4.3.1 Binary Thinning

Binary thinning is a so-called morphological operation [112]. Morphological operations are applied to binary images, e.g. images that are divided into a fore- and background, and change the morphology of the foreground. During the thinning algorithm, pixels are removed from the foreground based on the Euler characteristic of the connected components contained in the image. The Euler-characteristic of a (not necessarily convex) polygon with v vertices, e edges and f convex faces can be calculated with the Euler-Poincaré formula [75]

$$\chi(X) = v - e + f. \tag{4.9}$$

A convex polytope X_c always fulfills $v = e$ and $f = 1$, so $\chi(X_c) = 1$. A non-convex polytope will be first divided into convex sub-shapes and all vertices, edges and faces are counted separately and are then added. If a polytope has holes in it, $\chi <= 0$. Polytopes with the same topology, e.g. a donut and a cup, always have the same Euler characteristic. On a voxel grid all shapes can be defined as polytopes. Adding or removing a pixel that belongs to a connected component can alter its Euler-characteristic. Voxel configurations in a $2 \times 2 \times 2$ grid can be defined based on a chosen adjacency system that define vertices, edges and faces of the polytope defined by the non-zero voxels in a volume. Simply counting all occurrences of each of those configurations and adding their contribution in (4.9) yields the Euler characteristic for each connected component or the whole volume [12].

The thinning algorithm checks pixel configurations that do not alter the connectivity of connected components. The pixels that are part of these configurations can safely be removed without altering the Euler-characteristic.

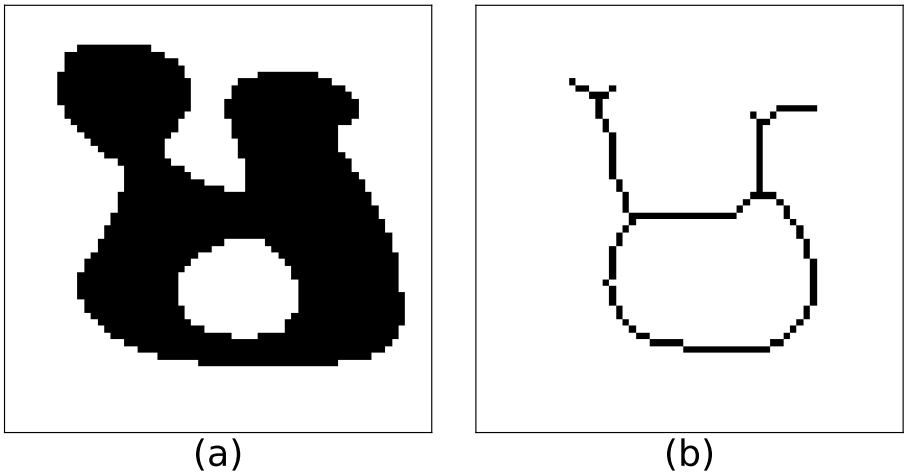


FIGURE 4.10: Binary object (a), with the black pixels indicating non-zero values, and its skeleton (b) computed using the thinning algorithm described in Subsection 4.3.1.

The algorithm iterates over the image until no more pixels can be removed from the image without changing the Euler characteristic. What is left is called the "skeleton" of the image. An example of skeletonization in the 2D case is shown in Figure 4.10.

In application to fibers, this means that the fiber voxels can be segmented and the medial axis of the connected components can be extracted in that way [72] as described above. The implementation [113] in the Insight Toolkit (ITK) [114] is used for computing the skeleton in this work.

4.4 Fiber Tracing

Working with curved fibers brings with it additional requirements when extracting individual fibers from a reconstruction. The fiber voxel extraction described in the previous section is only slightly adapted in this method compared to PARE, but the resulting voxel coordinates do not form straight lines anymore. This means the Hough transform would have to be adapted to

accommodate more parameters, while being limited by the sampling in the Hough accumulator space.

Such an increased parameter space would require several additional dimensions, rapidly increasing the storage space required, due to a phenomenon dubbed "the curse of dimensionality" [115]. It refers to the effect that if the dimensionality of a space increases, the volume of the space grows exponentially at a rate that makes most realistically possible sampling methods in a Hough parameter space sparse. This in turn significantly reduces the best possible accuracy that can be achieved with such a discrete space. A more flexible option is to "follow" the fiber center lines algorithmically, recording the visited voxels and chaining them together to a single fiber cluster. Such algorithms have been developed to map brain connectivity from diffusion Magnetic Resonance Images and are commonly referred to "tractography" algorithms.

4.4.1 Tractography

Tractography is a term for techniques that visualize the connectivity of nerves in the brain using local direction information. The direction information is derived from the diffusion tensor, that is obtained using diffusion-weighted Magnetic Resonance Imaging [107]. Diffusion MRI measures the magnetic resonance of molecules, enabled by the nuclear magnetic resonance (NMR) phenomenon [116]. The nuclei of atoms absorb electromagnetic radiation and subsequently re-emit radiation themselves - the response - when a static magnetic field is perturbed by an oscillating magnetic field. The frequency of that oscillation determines the kind of response and needs to be adjusted to the nuclei that are measured. Because human (and animal) tissue is filled with water, a useful nucleus to adjust to is hydrogen. Water is constantly moving in a process called Brownian motion. The molecules flow towards areas with lesser concentration of water and away from areas with high concentration. This movement is called diffusion. Inside the white matter fibers of the brain the diffusion is restricted by the geometry of the fibers. The flow rate is consequently highest along the direction of a white matter fiber bundle and lowest perpendicular to it. In a standard MRI, the response of the molecules is measured using multiple radio frequency pulses, called the *spin echo sequence*[117] and then interpreted. If the measurement is repeated after a small time interval, the so called diffusion time, and combined with a change in the static magnetic field, the movement of the water molecules cause the response to

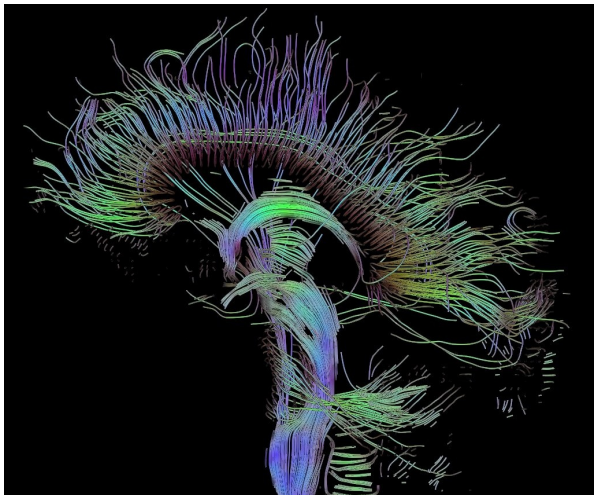


FIGURE 4.11: Tractography of a brain using a publicly available dataset from http://www.sci.utah.edu/~gk/DTI-data/. Visualization by Thomas Schultz, CC BY-SA 3.0, taken from wikipedia on 09.09.2022.

differ slightly between those evaluations [118]. From those differences the direction of the diffusion can be calculated.

With this direction information, the white matter fibers can be visualized. A simple approach is to send an imaginary particle from a random spot and let it "walk" with a constant speed, always following the diffusion tensor's main direction. Recording each location this particle "visits" results in a path that follows the fiber structures in the brain. Repeating this with multiple particles results in images like the one shown in Figure 4.11.

Fiber Assignment by Continuous Tracking (FACT) [119] is an algorithm that generates traces from the diffusion tensor in exactly that way. The diffusion directions are given by the Eigenvectors $\boldsymbol{v}_{\lambda_i}$ of the tensor. The relations between the Eigenvalues λ_1, λ_2 and λ_3 corresponding to each of those Eigenvectors indicate the "shape" of the diffusion. For example if $\lambda_1 \approx \lambda_2 \approx \lambda_3$, the diffusion is isotropic. Anisotropic diffusion along a cylindrical structure could be indicated by $\lambda_1 >> \lambda_2 \approx \lambda_3$, where the major direction is approximately aligned with λ_1.

The fiber traces are generated by starting at a random voxel in the MR image and moving a subvoxel-sized step into the direction indicated by the major axis of the diffusion tensor. This process is repeated until either the end

of the volume is hit, or until the stopping condition $R < 0.8$ is fulfilled. The value R is computed as the inner product of the principal direction vectors of a neighborhood around the current voxel:

$$R = \sum_{i=0}^{s} \sum_{j=0}^{s} \frac{||\boldsymbol{\nu}_{\lambda_{1i}} \cdot \boldsymbol{\nu}_{\lambda_{1j}}||}{s(s-1)}, \tag{4.10}$$

where $\boldsymbol{\nu}_{\lambda_{1i}}$ and $\boldsymbol{\nu}_{\lambda_{1j}}$ the normalized principal diffusion direction at one of the voxels and s is the number of voxels that are considered for this computation. The region considered is a 26-neighborhood ($3 \times 3 \times 3$) for the 3D case and a 8-neighborhood (3×3) for the 2D case. The value for R is large when the direction vectors are strongly aligned and small when they are not. Once a trace has stopped based on the stopping criterion, the algorithm is repeated in the opposite direction from the original starting point. Applying these steps for a number of random points in the volume reveals the structure of the connections of the nerves in the brain.

4.4.2 Fiber Assignment by Continuous Tracking for cuPARE

Applying this technique to reconstructions of GFRP requires some adjustment, as CT data does not provide local direction information. To acquire this additional information, the local direction can be computed using the Hessian matrix

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \frac{\partial^2 f}{\partial x_1 x_3} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 x_3} \\ \frac{\partial^2 f}{\partial x_3 x_1} & \frac{\partial^2 f}{\partial x_3 x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{pmatrix} = (\nabla \nabla^{\mathrm{T}})(f). \tag{4.11}$$

It represents the partial second derivatives of all coordinate directions, which can be used to estimate the local curvature in an image [61]. As described in [120], the fiber traces are found by computing the Hessian matrix for each voxel in the reconstructed volume and computing the Eigenvalues and Eigenvectors. The Eigenvector corresponding to the Eigenvalue with the smallest magnitude is the vector indicating the direction of lowest curvature [121]. That vector serves as a substitute for the the diffusion tensor's principal direction axis.

To be mainly sensitive to structures of the size of the analyzed fibers, the definition of the Hessian is slightly altered to accommodate a scale σ

$$H_\sigma = (\nabla\nabla^{\mathrm{T}})(f * g_\sigma). \tag{4.12}$$

Here the term g_σ represents a smoothing of the function f with a Gaussian kernel with standard deviation σ. The value for sigma is the fiber radius $\sigma = r_{\text{fiber}}$. With this smoothing, structures that are thinner than 2σ produce a significantly smaller response in the output image, effectively suppressing them. If multiple different fiber radii are present in the image, this operation can be repeated for each of those radii. In that case, the response at each voxel in the reconstruction is the maximum of all responses in the computed Hessian images. Applying the FACT algorithm without any changes results in traces that follow the directions of the fibers nicely. However, due to the smoothing the vector field spreads out in all directions, causing feathered out ends. A visualization of traces using the software tool MRTrix3 [122] is shown in Figure 4.12.

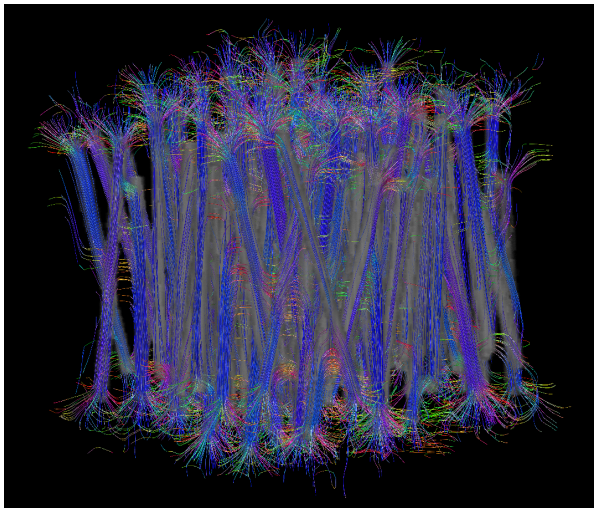


FIGURE 4.12: Result of applying the unchanged FACT algorithm to a simulated volume of straight fibers using the direction information obtained from its Hessian. It is clearly visible that the fiber direction can be approximated well, but that the traces feather out at either end of each fiber.

As FACT is designed as a visualization algorithm, the method needs to be adapted to generate only one trace per fiber and to not combine fibers that are

distinct objects in the volume. To that end, the fiber center lines are extracted and only the direction information in those voxels is used at first. A small, straight, cylindrical region is created around each starting voxel, with its axis aligned with the principal direction vector. From that cylinder segment on, the tracing continues to stack cylinder segments, jumping several voxels at the same time. The stopping criterion from (4.10) was found to not work well due to noise, so new criteria were added.

The tracing starts from a random point in the list of all detected fiber voxels, subsequently called seed points. The following procedure is repeated until all the seed points are assigned to a trace:

1. If there are seed points left, randomly select one of them, otherwise stop iterating;
2. A line segment with length $3r_f$ is centered around the point. Its direction is chosen from the local direction vector field at that point, linearly interpolated.
3. Find the closest other seed point to the end point of the line segment.
4. This point will be the next point in the tracing if the two consecutive line segments meet the following criteria
   (a) The angle between the direction vectors of the two line segments is $< \theta_1$
   (b) The distance of the two closest points on the line segments is $< r_f$
   (c) The angle between the current line segment's direction vector and the connection line through the two closest points is $< \theta_2$.

   Here, r_f is the radius of the fibers and the angles $\theta_1 = 10\,^\circ$, and $\theta_2 = 20\,^\circ$ were found to give good results.
5. If the conditions are met, remove all seed points from the list that would be located inside a cylinder with the line segment as its central axis and radius r_f. Then go to step 2. If the condition is not met, stop the tracing and go to step 6;
6. Start the tracing from the same initial point chosen in step 1, but in the opposite direction, as it is done by Teßmann et al.[123]. Flip all vectors in the tracing in the same way; If this is the second time going through this step, go to step 1;

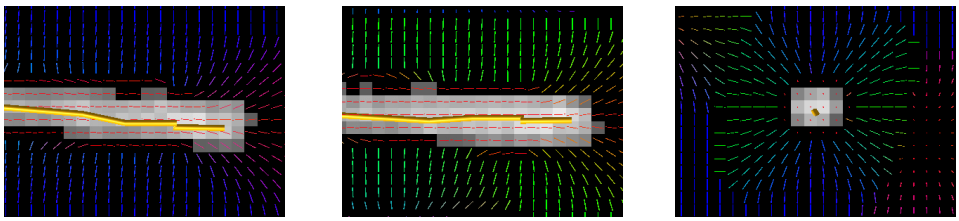


FIGURE 4.13: Slices through a vector field obtained using the described Hessian processing. In light gray the fiber voxels are shown, superimposed on that is the vector field. The color of the vectors corresponds to their x, y, z components encoded as RGB, respectively. Overlaid in yellow is the trace line rendered as a tube. Figure reproduced from [120], © Society of Photo-Optical Instrumentation Engineers (SPIE).

The result of the algorithm is a list of sets of 3D coordinates that describe each individual fiber trace that was found. Some of these traces might belong to the same fiber, but are initially detected as separate. To combine those segments a final clustering is applied, where fiber traces that are intersecting or touching are combined into one fiber segment. This step uses the same criteria as in item 4, but on the larger segments instead of the local direction vectors. The merging of the fiber traces is repeated until no more touching or intersecting fiber segments are found. A visualization of a local direction vector field around a simulated fiber is shown in Figure 4.13, with the resulting traced line highlighted.

This procedure is fully automatic, as opposed to semi-automatic methods like MetaTracts [100], which use different software packages for the individual sub-tasks such as tracing and clustering.

4.5 Optimization of Curved Fibers

With the curved fibers clustered by their center line coordinates, the next step is to convert the chain of points into a concise representation that can be adjusted easily based on few parameters as described in Section 4.2. For cuPARE the chosen representation is based on polynomials, which are well understood and easily adjustable in terms of the number of parameters. This means that both complex curvatures as well as simple arches can be represented without having to change representation. An example of a fiber curve

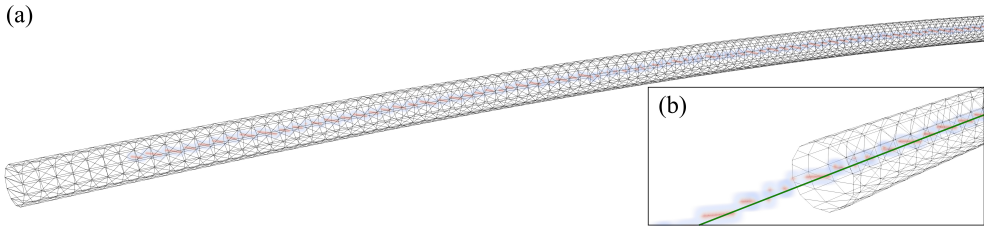


FIGURE 4.14: (a) Curve representing a fiber in 3d space and the voxels it was estimated from. A wire frame around the curve is shown to visualize the triangle mesh model that is used to represent the fiber for generating projections. (b) shows a close up of a part of the voxels (light red and blue), center line (dark green). The wire frame (black) is partially cut to reveal the voxels and center line. Figure reproduced from [106] under CC 4.0.

based on its voxel coordinates is shown in Figure 4.14. The wireframe surrounding the center line is a representation of a mesh model generated from the model. This model is used in a later processing step of the algorithm to generate fiber projections.

After the tracing step the processing could be considered finished, as the fibers have been detected and parameterized. However, like in the PARE algorithm this representation can still be made more accurate by applying an optimization step. The PARE algorithm used gradient descent to this end, but experiments with the curved fibers showed that gradient descent was not suited well for this adapted algorithm. The gradient information, which a lot of optimization algorithms use, is hard to estimate from finite differences. Therefore the choice fell on a numerical optimization method that works without derivatives - the Nelder-Mead Simplex algorithm [124].

4.5.1 Nelder-Mead Simplex Algorithm

The Nelder-Mead Simplex algorithm is an unconstrained optimization algorithm. As most optimization algorithms, it requires an objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, which maps an n-dimensional parameter vector to a scalar. The algorithm gets its name from the way it evaluates the objective function. A simplex is a generalization of a triangle in 2D that connects $n + 1$ vertices in an n-dimensional space, such that the faces form the convex hull of all

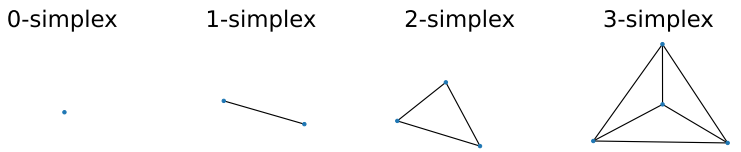


FIGURE 4.15: Examples of simplices in 0, 1, 2 and 3 dimensional space.

vertices. The first four simplices: a point, line segment, triangle and a tetrahedron, are visualized in Figure 4.15. A simplex is called a regular simplex if the connections of the vertices of the convex hull are all the same length.

Before the start of the iterations, a simplex, denoted with $\triangle_S$, with a side length of δ is generated and the objective function at the $n+1$ vertices is evaluated.

The method iteratively moves the vertices of the simplex to approximate the optimimum of the objective function f. The first step is to sort the vertices of the simplex $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n+1}$ from smallest to highest objective function value

$$f(\boldsymbol{x}_1) \leq f(\boldsymbol{x}_2) \leq \cdots \leq f(\boldsymbol{x}_{n+1}). \tag{4.13}$$

Vertex $\boldsymbol{x}_1$ is called the *best vertex* and consequently the last vertex in the order is called the *worst vertex*. From this state the algorithm uses one of four possible operations, depending on the objective function values of the vertices.

1. *reflection*, associated with the parameter α,
2. *expansion*, associated with the parameter β,
3. *contraction*, associated with the parameter γ,
4. *shrink*, associated with the parameter η.

The parameters are used to compute new vertices for the simplex as shown in Algorithm 4.

Algorithm 4 Nelder-Mead simplex algorithm

1: evaluate $f(\boldsymbol{x}_i) \forall \boldsymbol{x}_i \in \triangle_S$ and sort so that (4.13) holds.
2: **Reflection:**
3: compute reflection point $\boldsymbol{x}_r = \bar{\boldsymbol{x}} + \alpha(\bar{\boldsymbol{x}} - \boldsymbol{x}_{n+1})$.
4: **if** $f(\boldsymbol{x}_1) < f(\boldsymbol{x}_r) < f(\boldsymbol{x}_n)$ **then**
5: $\quad \boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}_r$
6: **end if**
7: **Expansion:**
8: **if** $f(\boldsymbol{x}_r) < f(\boldsymbol{x}_1)$ **then**
9: $\quad$ compute expansion point $\boldsymbol{x}_e = \bar{\boldsymbol{x}} + \beta(\boldsymbol{x}_r - \bar{\boldsymbol{x}})$.
10: $\quad \boldsymbol{x}_{n+1} \leftarrow \begin{cases} \boldsymbol{x}_e & \text{if } f(\boldsymbol{x}_e) < f(\boldsymbol{x}_r) \\ \boldsymbol{x}_r & \text{otherwise} \end{cases}$
11: **end if**
12: **Outside Contraction:**
13: **if** $f(\boldsymbol{x}_n) < f(\boldsymbol{x}_r) < f(\boldsymbol{x}_{n+1})$ **then**
14: $\quad$ compute contraction point $\boldsymbol{x}_{oc} = \bar{\boldsymbol{x}} + \gamma(\boldsymbol{x}_r - \bar{\boldsymbol{x}})$
15: $\quad$ **if** $f(\boldsymbol{x}_{oc}) < f(\boldsymbol{x}_r)$ **then**
16: $\quad\quad \boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}_{oc}$
17: $\quad$ **else**
18: $\quad\quad$ go to line 28
19: $\quad$ **end if**
20: **end if**
21: **Inside Contraction:**
22: **if** $f(\boldsymbol{x}_r) \geq f(\boldsymbol{x}_{n+1})$ **then**
23: $\quad$ compute contraction point $\boldsymbol{x}_{ic} = \bar{\boldsymbol{x}} - \gamma(\boldsymbol{x}_r - \bar{\boldsymbol{x}})$
24: $\quad$ **if** $f(\boldsymbol{x}_{ic}) < f(\boldsymbol{x}_{n+1})$ **then**
25: $\quad\quad \boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}_{ic}$
26: $\quad$ **end if**
27: **end if**
28: **Shrink:**
29: **for** $2 \leq i \leq n+1$ **do**
30: $\quad \boldsymbol{x}_i \leftarrow \boldsymbol{x}_1 + \eta(\boldsymbol{x}_i - \boldsymbol{x}_1)$
31: **end for**

4.5.2 Application of Nelder-Mead to Curved Fibers

To keep the optimization parameter space as low dimensional as possible, each individual fiber is optimized separately. Thus, for a single fiber parameter set the optimization problem becomes

$$\hat{\xi}_j = \arg \min_{\xi_j \in \mathbb{R}^{3(k+1)}} f_{\text{loss}}(\boldsymbol{p}, \boldsymbol{W}\boldsymbol{x}_v(\xi_j)), \tag{4.14}$$

where f_{loss} is the chosen objective function, k the degree of the polynomials representing the fiber center line, $\boldsymbol{p}$ the measured projection data and $\boldsymbol{W}\boldsymbol{x}_v$ the forward projection of an estimate of the reconstruction as a function of the fiber parameter list ξ_i.

The parameter values are converted to be in the range of $[0, 1]$ while they are manipulated during optimization, for improved numerical stability. To convert to the normalized range, the whole population of estimated fibers is considered. The conversion is therefore mapping 0 to the lowest value that each coefficient had in the population and 1 to the highest value in each coefficient. This keeps the step-size the same for each parameter, even when the parameters had vastly different ranges originally. Once a fiber model needs to be projected, the values are converted back to their original range.

With the more complex fiber model, the objective function becomes increasingly complex as well. The mean of the sum of the squared projection differences or root mean squared error (RMSE) is notoriously sensitive to slight differences in intensity. As the main objective is to estimate the fiber shape, the Structural Similarity Index Loss (SSIM Loss) [125]

$$\begin{aligned} \text{SSIM}_{\text{loss}}(\boldsymbol{A}, \boldsymbol{B}) &= \frac{1}{2}\left(1 - \frac{1}{M}\sum_{i=1}^{M} \text{SSIM}(\boldsymbol{a}_{\text{i}}, \boldsymbol{b}_{\text{i}})\right) \\ &= \frac{1}{2}\left(1 - \text{MSSIM}(\boldsymbol{A}, \boldsymbol{B})\right). \end{aligned} \tag{4.15}$$

was chosen as the objective function for the optimization. The SSIM measure is computed on M local windows $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$, which are corresponding regions in the projections $\boldsymbol{A}$ and $\boldsymbol{B}$ and then averaged resulting in a scalar measure. This measure is widely used in Deep Learning in combination with the RMSE to tackle similar problems in image similarity assessment and has proven to be much more robust as a measure than the RMSE in our experiments. To compensate for noise, the projections can be smoothed with a Gaussian filter of variable width, depending on the noise level of the input images. In

addition to the SSIM Loss, we also introduce a penalty term which keeps the fiber that is currently being optimized within the bounding box Ω of the reconstructed volume. Due to the representation as a polynomial on the interval $[0, 1]$, the minimum and maximum coordinates can be obtained from $g(u = 0)$ and $g(u = 1)$. The term $g(0)$ corresponds to the constant coefficients of g and $g(1)$ to the sum of all coefficients. With the conditions

$$\Omega_{\min} \leq g(0) \leq \Omega_{\max} \tag{4.16}$$

and

$$\Omega_{\min} \leq g(1) \leq \Omega_{\max}. \tag{4.17}$$

the total penalty added to the SSIM Loss becomes

$$\begin{aligned} p(g, m) = & \, mL(\Omega_{\min} - g(0)) \\ & + mL(\Omega_{\min} - g(1)) \\ & + mL(g(0) - \Omega_{\max}) \\ & + mL(g(1) - \Omega_{\max}), \end{aligned} \tag{4.18}$$

where m is the current iteration of the optimization, meaning that the penalty for the same parameter values increases over time. The function L is the logistic function

$$L(x) = \frac{1}{\mathrm{e}^{-10^4 x}}. \tag{4.19}$$

The constant -10^4 was determined empirically and serves to sharpen the transition between the penalty and non-penalty regions compared to the unmodified logistic function.

4.5.3 Projection of Mesh Models

In our previous work [77], fiber voxel models were generated from their parametric descriptions and subsequently the voxels were projected using the ASTRA toolbox [47]. While this approach is simple and effective, it requires expensive computations both for generating and projecting the voxels. In this work, a triangle mesh representation based on the parametric fiber descriptions is used to project the fibers. That mesh model is generated using the Visualization Toolkit (VTK) [1] tube filter [126] on the detected fiber center lines. To achieve realistic volume borders, the tube model is cut off at

[1] using VTK version 9.1

the volume boundary using Boolean operations on the meshes [127], if it is protruding out of said volume. The resulting fiber mesh is projected using a custom, GPU accelerated, mesh projector [128], which reduces the computational load significantly.

4.5.4 Reduction of Computational Cost

The projections for the optimization can be the ones used for reconstruction of the volume, a subset thereof or even a small set of higher resolution projections acquired separately. Additionally, the projections can be cropped during the optimization of an individual fiber to reduce the computational complexity of computing the objective function (4.15). To that end, the axis aligned bounding box of the current fiber being optimized is computed in reconstruction space. The bounding box is then projected onto the detector for each projection angle and the largest overlapping region of all projected bounding boxes is determined. Finally, the region is extended in its largest dimension to span the whole width or height of the detector. As a result, the memory footprint can be reduced significantly, depending on the orientation and position of the fiber relative to the rotation axis of the tomography setup. Depending on the fiber orientation, this reduces the amount of data to be transferred from and to the GPU.

4.5.5 Computational Complexity

The algorithm presented here comes at a high computational cost due to the amount of data contained in a three dimensional image. It is therefore important to implement the individual operations efficiently. The reconstruction, which is a pre-requisite, is not considered in this section. The purely voxel based image operations like thresholding and the removal of voxels with a certain amount of non-zero neighbors, have a time complexity of $\mathcal{O}(n_\mathrm{r})$, where n_r is the number of voxels in the reconstruction. The most computationally expensive operation in the fiber voxel detection pipeline (see Subsection 4.3) is the thinning operation. Binary thinning is achieved by iteratively removing voxels that do not alter the topology of the connected components, based on pre-defined pixel configurations, as described in Subsection 4.3.1. The time complexity for the thinning/skeletonization is therefore $\mathcal{O}(n_\mathrm{r} \cdot i)$, where i is the number of iterations needed, which is assumed to be $i \ll n_\mathrm{r}$. It is possible to parallelize this step at the cost of guaranteed topological preservation, which has not been done in this work. The Nelder-Mead optimization

algorithm has a time complexity of $\mathcal{O}(n \cdot j)$, with the simplex transformation taking n operations and j the number of iterations until convergence [129]. During the optimization, the objective function is evaluated multiple times. It depends on the number of pixels in all projections n_p as well as the size of local window m_p used in the computation of the SSIM loss, $\mathcal{O}(n_\mathrm{p} \cdot m_\mathrm{p})$. The projections that are simulated for the comparison with the measured projections can be computed in $\mathcal{O}(n_\mathrm{f} \cdot n_\mathrm{p})$ time in the worst case, where n_f is the number of faces in the mesh model. This depends on the relative size of the mesh's faces compared to the detector pixels. Assuming the computation of the objective function to be the most costly operation in the optimisation, the time complexity can be assumed to be $\mathcal{O}(n_\mathrm{f} \cdot n_\mathrm{p} \cdot m_\mathrm{p} \cdot j)$ or lower. The meshes of the fibres that are not changed, can be pre-computed from the remaining mesh models, adding an additional one-time overhead. Assuming the operations in the simplex transformation to be negligible compared to the computation of the objective function, the total complexity of cuPARE optimization can be assumed to be at least $\mathcal{O}(n_\mathrm{f} \cdot n_\mathrm{p} \cdot m_\mathrm{p})$.

4.6 Experiments and Results

The cuPARE algorithm was tested with both simulated and real data. The simulations were performed in the same way as in Subsection 3.5.1, using RSA to accumulate fibers and the ASTRA toolbox to simulate the projections. The cuPARE algorithm was then applied to a realistic, publicly available data set of unidirectional glass fibers embedded in a polymer resin (XCT_H) [99], which was already used in Subsection 4.2.4.

For the optimization of the simulated dataset the stop condition for the optimization of each fiber was either after 1000 iterations, or if the simplex vertex with the best and the simplex vertex with the worst value in f_loss had at most a difference of 1×10^{-7}.

At first the method was verified using simulated data. Like in Subsection 3.5.1, a fiber composite phantom was simulated consisting of a polymer matrix and glass fibers. The RSA algorithm was again used to position fiber shapes into a volume in random orientations following a VMF distribution (see (3.22)) in $\mathbb{R}^3$ with the mean direction $\boldsymbol{\mu} = (1, 0, 0)^\mathrm{T}$.

Instead of creating straight fibers, the fibers are generated with curvature using a parabola fit. To that end, a straight fiber is created and its center point is then moved in a random direction perpendicular to the main axis by

a random amount. Then a parabola is fitted through these three points, yielding the center line of the generated fiber. This displacement, influencing the amount of curvature, was randomly chosen from the interval $[6, 14]$ voxels.

A simulated reconstruction of the generated dataset, which will be referred to as *Sim*, is shown in Figure 4.16. Here, 186 fibers were added to a volume of size $400 \times 400 \times 400$ voxels with varying amounts of curvature. The fiber radius was (9.75 ± 0.45) µm (6.5 ± 0.3 voxels) with a voxel size of 1.5 µm, simulating slight variations in the fiber production process. The spread parameter κ_A was arbitrarily set to 50, modeling nearly uni-directional fibers.

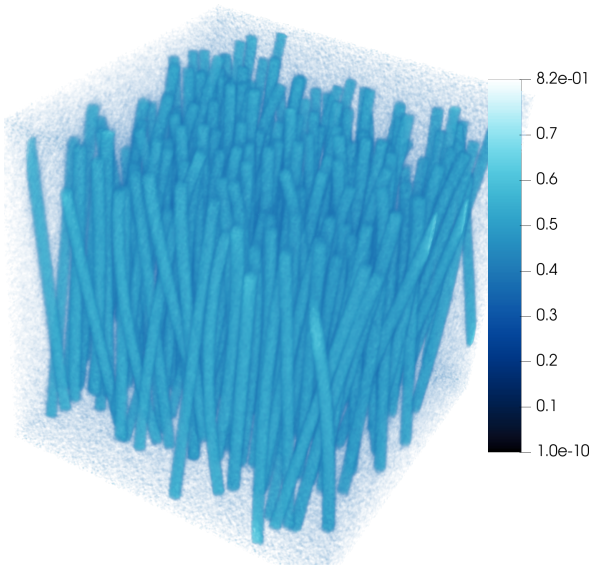


FIGURE 4.16: Reconstructions with $200 \times 200 \times 200$ voxels of the simulated datasets. The reconstruction was performed using 100 iterations of the Barzilai-Borwein [46] algorithm and the background was made transparent to make the fibers more visible. Note that the fibers are oriented along the X-axis. Figure reproduced from [106].

To simulate a realistic experiment, the projection geometry for the simulated datasets was modeled after the scanner setup in our laboratory [8]. The source-object distance (SOD) was 19.9 mm and the source-detector distance (SDD) 1795.0 mm and a 1024×1024 detector with a pixel size of 75 µm. This geometry was then used to project the $400 \times 400 \times 400$ phantom with 200 equidistant projections distributed on the full circle.

4.6.1 Volume Overlap Measures

The quality of the reconstructed fiber model of the simulation experiments was evaluated based on the generated ground truth data. This was done by means of a volumetric overlap measure introduced by Fröhler et al. [130]. In Subsection 4.2.4 the *dminavg* and *dmaxavg* measures were used to assess the quality of fit for the curve estimation. Distance based measures are well suited for that task, however matching pairs between two sets of fibers, for example a ground truth and an estimate, they often do not result in the match that a human observer would have assigned. The issue here is that fibers that are close in distance can be fibers running parallel to the fiber that is the "true" match, while that "true" match has some kind of error that increases the distance of some of the fiber points relative to its match in the other set. This happens when one of the fibers is shorter than the other, or merges with a neighboring fiber due to a tracing error. Additionally the measures are directional, meaning that the choice of "reference" fiber influences the end result of the computation, as indicated by Figure 4.17a.

To solve these issues, overlap based measures were introduced in [130], which approximate a match that a human observer would assign. Each fiber is approximated by a sequence of cylinders with a fixed radius and length. A fixed number of points is sampled within one of the fibers to be evaluated. Then it is checked for each of those points whether or not that point lies within the other fiber. The number of points sampled is equal in each straight segment and depends on the volume quotient of the entire fiber and the segment. That way even sampling over the entire fiber is guaranteed. Whenever there is a transition from one segment to the next, there is a small section that could be sampled twice, once in each segment and a small section that is not sampled at all. This is visually explained in Figure 4.17c. With low curvature these volume sections are small, thus the error introduced by this is negligible, akin to the error the spherical end caps introduced in the collision detection in Subsection 3.5.2.

The number of points contained in both fibers is an approximate measure for the overlapping volume between the two fibers. With the total fiber volumes the overlap measure can be computed as

$$d_3^o(f_a, f_b) = 1 - \mathbb{O}(f_a, f_b) \begin{cases} \frac{V(f_a)}{V(f_b)} & \text{if } V(f_a) < V(f_b) \\ \frac{V(f_b)}{V(f_a)} & \text{otherwise} \end{cases}, \tag{4.20}$$

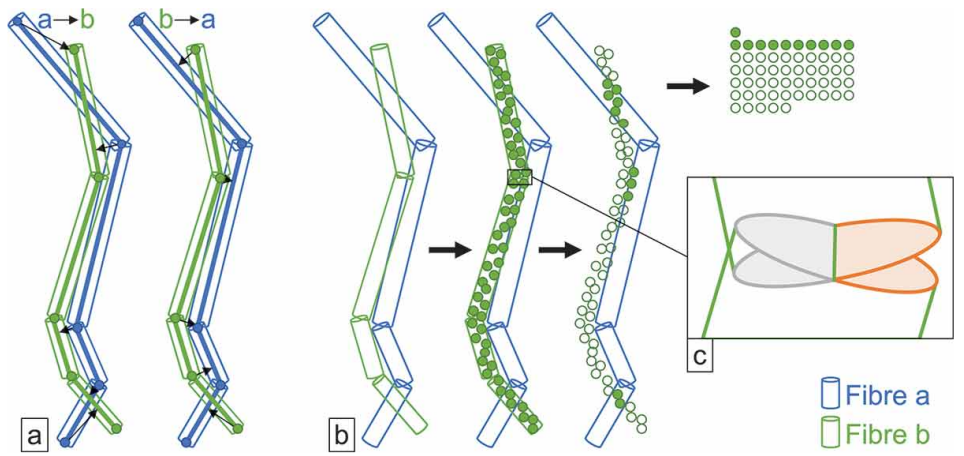


FIGURE 4.17: Visualization of how the distance based (a) and overlap based (b) measures are computed. Subfigure (c) shows the small region of error introduced by every transition between straight segments. Figure reproduced from [130] under CC 4.0.

where $\mathcal{O}(f_a, f_b)$ denotes the volumetric overlap of two fibers f_a and f_b, and $V(\bullet)$ denotes the volume of a fiber.

If a mesh model and a continuously defined center line are available, as is the case in cuPARE, the computation of the overlap measures can be made more precise by considering the relation between the volume of a cylindrical tube following the center line and the volume of the intersection $f_a \cap f_b$ of the two fibers f_a and f_b. The intersection can be computed using a Boolean operation on the two meshes using a package such as PyMesh [127], as shown in Figure 4.18. This more precise computation is used when the actual quality of the match is supposed to be evaluated, as opposed to the matching fiber in two sets. The former is computationally more expensive, the latter less precise.

4.6.2 Variation of Noise Level

An obvious first assessment is the performance of a method in the presence of noise. Therefore, Poisson noise was added to the projection data, varying the simulated photon count per detector pixel from 250 photons up to 5000

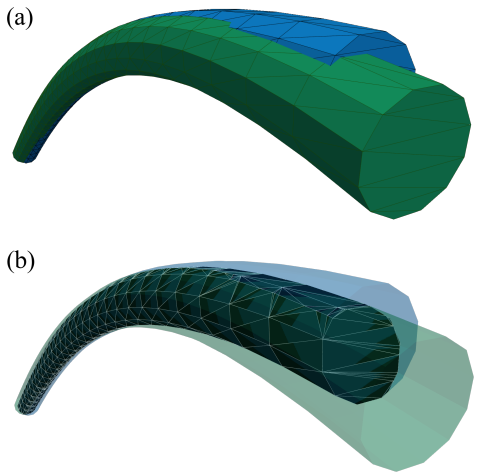


FIGURE 4.18: Visualization of two partially overlapping fiber meshes and their common volume. In subfigure (a) the green and blue meshes are shown solid, in (b) the same meshes were made transparent and in dark gray the mesh resulting from the Boolean intersection of the two meshes is shown.

photons. Additionally an intensity drop-off from the center due to the X-rays hitting the detector at an oblique angle as well as the increased distance due to the flatness of the detector according to Lambert's cosine law [87] was simulated. Finally, a point spread function [87] was simulated by applying a Gaussian filter with a standard deviation of $\sigma = 0.3$ pixels.

To simulate a lower resolution scan, the "binned" by a factor of 2. Binning is the operation of combining several pixel values into one "bin", which gets assigned the average values of the pixels it contains. A binning factor of n in this context means that a section of $n \times n$ pixels is averaged and treated as a single pixel the new, binned image. These lower resolution images are used for reconstruction in the further experiments, the higher resolution images are used for the optimization procedure.

Reconstruction from the projections was performed using 100 iterations of the BB reconstruction algorithm with a lower bound of 0. This resulted in a voxel grid of $200 \times 200 \times 200$ which can be analyzed by the cuPARE algorithm.

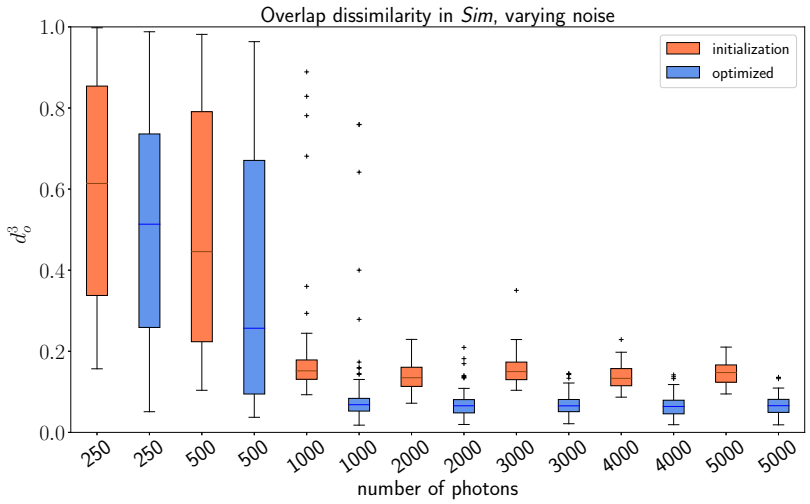


FIGURE 4.19: Boxplot of the *overlap dissimilarity* values for each fiber in *Sim* before and after optimization relative to the ground truth. Orange boxplots show the estimated fibers compared to the ground truth, blue boxplots show the same fibers after being optimised. Figure reproduced from [106].

The peak signal-to-noise ratio (PSNR)

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{\max(\boldsymbol{p}_{\mathrm{gt}})}{\mathrm{MSE}(\boldsymbol{p}_{\mathrm{gt}}, \boldsymbol{p}_{\mathrm{noisy}})} \right) \tag{4.21}$$

of the simulated projections is reported in Table 4.1. Here $\boldsymbol{p}_{\mathrm{gt}}$ is the noiseless projection and $\boldsymbol{p}_{\mathrm{noisy}}$ the projection with added Poisson noise. The function $\mathrm{MSE}(\bullet)$ denotes the mean squared error.

The result of the fiber detection quality as well as the quality of the fiber models after optimization are shown in Figure 4.19. It is clear that cuPARE is robust with respect to the amount of noise in the reconstruction, given a minimum sufficiently high PSNR/photon count. This is due to the fiber detection failing when the photon count drops below the threshold of 1000 photons. Improvement during optimization largely consists of the detection getting better with increasing PSNR, so the number of outliers declines as a function of the amount of noise. The median value of the *overlap dissimilarities* is

PSNR of the datasets in the noise experiment

Photons per pixel	PSNR low res [dB]	PSNR high res [dB]
250	26.22	26.24
500	29.25	29.29
1000	32.23	32.32
2000	35.15	35.33
3000	36.83	37.10
4000	37.98	38.35
5000	38.86	39.32

TABLE 4.1: PSNR (computed using (4.21)) values of the projections of the datasets with varying noise levels. Both the PSNR of the projections used for reconstruction (low resolution, middle column) as well as the projections used for optimization (high resolution, rightmost column) are reported. The leftmost column indicates the number of simulated photons per detector pixel.

essentially constant from 1000 photons per pixel, while the spread of the distribution decreases slightly and the number of outliers reduces greatly. The failure of the detection step with the low PSNR value exhibits itself in the fibers being split into many smaller pieces, which individually are aligned well with the fiber direction, but cannot be properly clustered in the tracing step. Slight mistakes in the fiber tracing can however be compensated through the optimization.

4.6.3 Variation of Number of Projections

A second important aspect of the optimization procedure is the amount of available information in the projection space, i.e., the number of projections. With more projections available, the accuracy of the optimization feedback loop increases, but the computational complexity increases as well. To find a good balance between those two factors, cuPARE was used to optimize the same detected fibers in *Sim* with varying amounts of projection images. A subset of 2 to 13 projections, equidistantly spaced on the circle was chosen for the experiments. To reduce the influence of noise, the fibers detected in the dataset with the largest PSNR from the previous experiment was chosen as the starting point for the optimization.

After optimization, the *dissimilarity overlap* was computed for the resulting fiber populations. Figure 4.20 shows the boxplots of the resulting *overlap dissimilarities* before and after optimization.

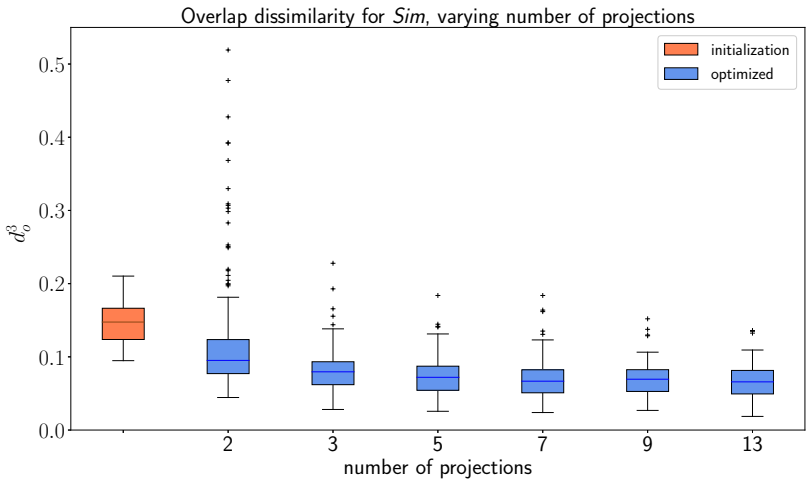


FIGURE 4.20: Boxplots of the *overlap dissimilarity* values after optimizing the fibers obtained from the dataset with 5000 simulated photons per pixel. The orange boxplot shows the match before optimization, the blue boxplots show the influence of the number of projections available to the optimizer. Figure reproduced from [106].

At first glance, the optimization succeeds in improving the fiber model accuracy even with only 2 projections available, with the improvement in the form of a reduction of the *overlap dissimilarity*. On further evaluation of the boxplots, the outliers show that some of the fibers move further away from their ground truth counterparts in terms of *overlap dissimilarity* for both 2 and 3 available projections. From 5 projections until 13, the improvement is apparent versus the initialization.

4.6.4 Variation of Curvature

Finally the curvature of the fibers was investigated. To that end, five additional datasets, shown in 4.21, with increasing curvature were generated.

The interval for the curvature offset described in Section 4.6 is defined by its center and a variation. For example, the interval [6, 14] is generated by a center value 10 and the variation value 4.

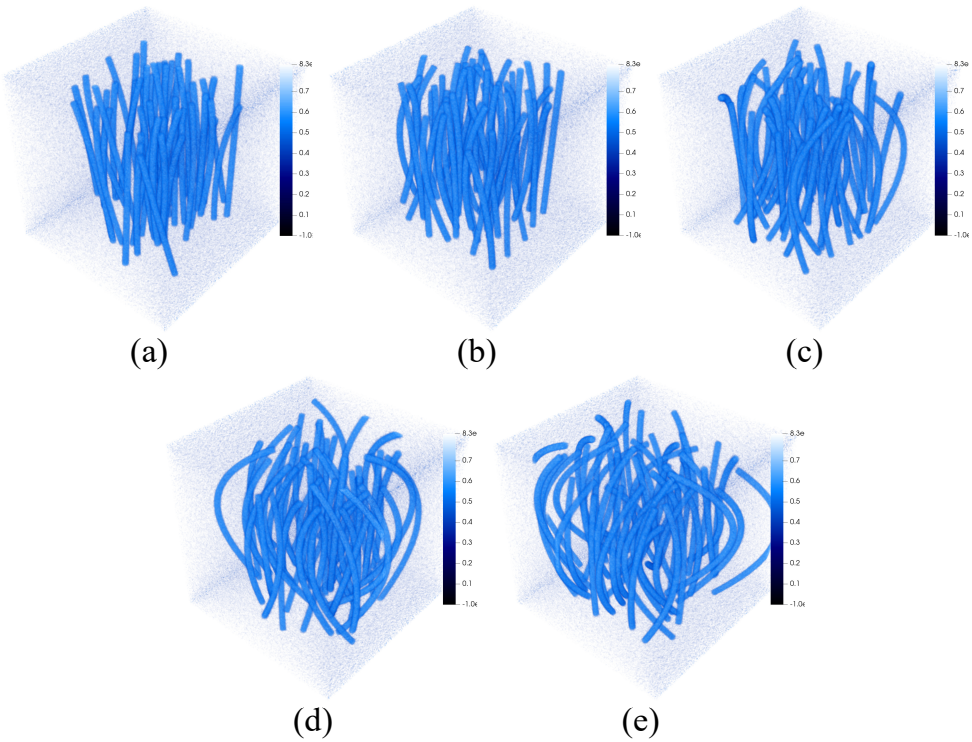


FIGURE 4.21: Reconstructions of simulated datasets containing fibers with varying curvature. The offset to generate the arch for the curved fiber was varied between 10 voxels (subfigure (a)) and 90 voxels (subfigure (e)) in steps of 20 voxels. The estimation quality of the fibers is shown in Figure 4.22. Figure reproduced from [106].

Keeping the variation the same, only the center value was varied for this experiment. The lowest curvature offset, 10 voxels, was the curvature offset chosen in the earlier experiments for *Sim*. Additionally the values 30, 50, 70 and 90 voxels were used. The corresponding datasets cover more volume because of the increased curvature, so only a small region in the center of the volume was chosen to make sure most fibers stay within the volume bounds. With increasing curvature, the initial estimates' *overlap dissimilarity* scores become worse.

As with the other experiments, the cuPARE algorithm was then used to

extract the fiber information, varying the parameters to fit the different curvatures. The extracted fibers were then optimized; the results of this optimization are shown in Figure 4.22.

The optimization procedure yields around the same median *overlap dissimilarity* 0.07 ± 0.02 for each dataset, however the number of outliers and the spread of the distribution increases with a larger curvature. The increase in outliers can be explained by the fact that more and more fibers are cut off by the volume boundaries. This effect is clearly visible in Figure 4.21.

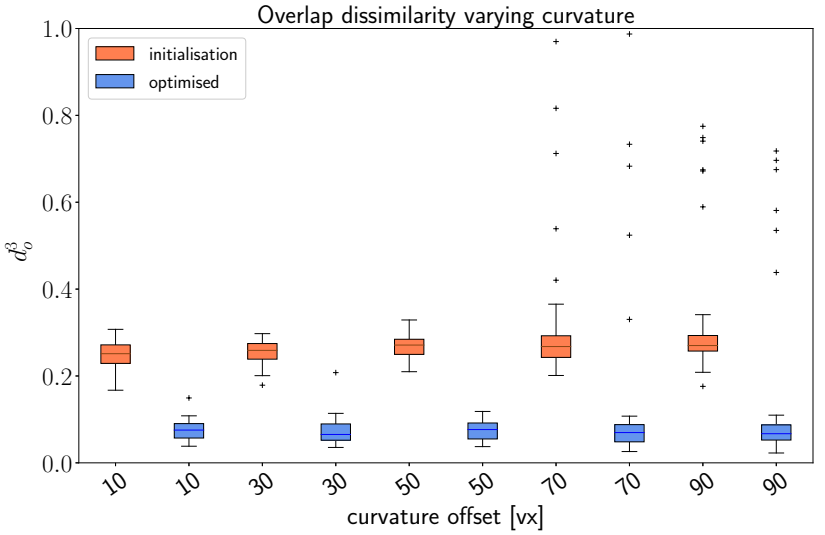
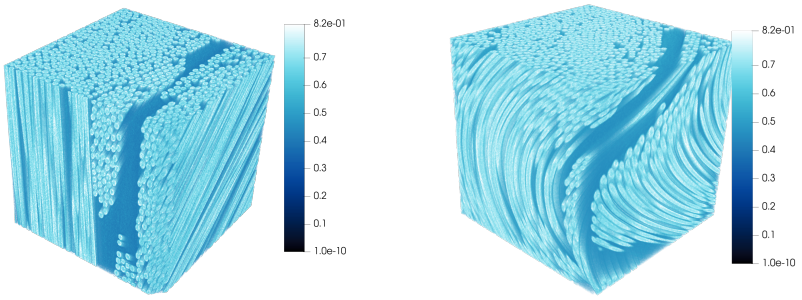


FIGURE 4.22: Boxplots of the *overlap dissimilarity* values when varying the curvature of the fibers in the datasets. Each indicated curvature offset corresponds to the displacement of the central point of the arch by that many voxels in a random direction, as outlined in Subsection 4.6.4. Figure reproduced from [106]

4.6.5 Real Data

As a final experiment, the cuPARE method was applied to a publicly available dataset of high resolution scans of a uni-directional glass fiber-reinforced polymer sample reconstructed from 4201 projections [99]. Due to the original projections used for the reconstruction of the dataset not being available, an experiment was simulated. To that end, a region of $500 \times 500 \times 500$ voxels



(A) (B)

FIGURE 4.23: Cutout of a region of 500 × 500 × 500 voxels from the high resolution dataset from [99] used as a realistic phantom after downsampling to 250 × 250 × 250 voxels. The intensities in the reconstruction were normalized to a range of [0, 1] and the polymer matrix (dark blue) was made semi-transparent to make the fibers more visible. Every voxel under 0.4 intensity was made invisible. Subfigure (a) shows the cutout unaltered, Subfigure (b) shows the same cutout region after applying a "twisting" affine transform to simulate stronger curvature in the fibers. Figure reproduced from [106].

was cut out of the high resolution X-ray CT dataset (named $\mathrm{XCT_H}$ in [99]). Using the ASTRA toolbox [47] 200 equiangular cone-beam projections, distributed over 360 degrees, were simulated from a downsampled version of that cutout with a size of 250 × 250 × 250 voxels as a phantom. This downsampled phantom is shown in Figure 4.23a. In what follows, this dataset will be referred to as XCT_H.

The fibers were extracted from a BB reconstruction with 100 iterations using the processing pipeline described above. Without a ground truth available for the fiber models, two additional fiber extraction methods were applied to this data and all 3 sets of detected fibers were used with the optimization framework described in Section 4.5. In total, the volume contains between 750 and 1000 fibers or pieces of fibers. The true number of contained fibers is unknown; each method detected a different amount of fibers.

For easier handling of the data the intensities of all datasets are expressed normalized, that is within the range $[0, 1]$, where 1 is the maximum intensity of the used datatype and 0 is the minimum intensity. The intensity of the polymer matrix was estimated at 0.04 ± 0.02 and the intensity of the fibers

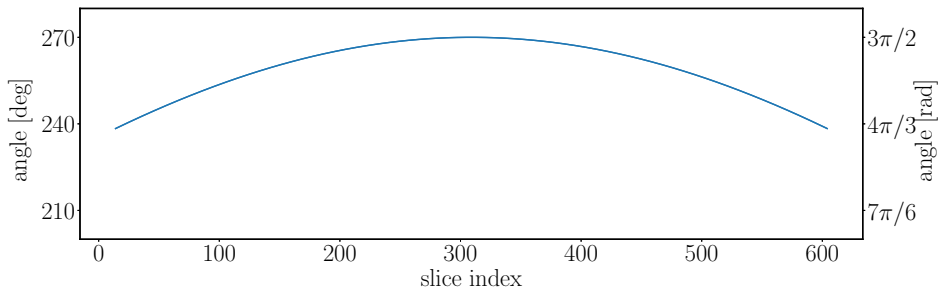


FIGURE 4.24: Relation between the slice index of the slices of the dataset shown in Figure 4.23b and the rotation angle of the image. The index 0 corresponds to the lowest Z-coordinate, the highest index to the highest Z-coordinate.

was estimated at 0.61 ± 0.22.

The geometry to simulate those projections was defined with an source-detector distance (SDD) of 55 mm and an source-object distance (SOD) of 10 mm. The voxel size was set to 1 µm and the pixel size of the 1024×1024 detector was 2 µm. This reduces the effective pixel size to 0.36 µm. This small effective pixel size does not affect the resolution of the simulated projections, as the voxels that are projected using ASTRA are of a fixed size of 1 µm, as described in [99]. During the optimization in the last step of the experiment, the fiber models do benefit from the smaller pixel size because they are defined by their vertices, whose position in space is defined with floating point precision. The vertex coordinates are encoded as multiples of the voxel size.

A second experiment with curved fibers was set up by applying a rotation to the slices of the full sized reconstruction with the angles varying with the shape of a Gaussian with an amplitude of 270 °, such that the images close to the central slice were more rotated than the images towards the extremities. The relation between the slice index starting from the bottom slice as index 0 is shown in Figure 4.24.

This produces a phantom containing curved fibers, as can be seen in Figure 4.23b. The fiber estimates generated by InsegtFibre were transformed in the same way, to get a reference estimation.

The different methods yielded varied results in the fiber detection step on the straight fiber volume. cuPARE detected 1034 fibers or fiber fragments, the InsegtFibre method found 766 and using the Fiber Characterization Pipeline "Template Matching" (FCP-TM) of Zauner et al. [73] 856 fibers and fiber fragments were identified. The fibers detected by InsegtFibre were found on the

original reconstruction XCT_H at full size and then cropped to the investigated region. The true number of fibers contained in the investigated volume is unknown. Both FCP-TM and cuPARE were applied to the downsampled data. A comparison of the reconstruction with renderings of the fiber estimates from each method are shown in Figure 4.25.

The individual optimized fibers are shown in separate plots to show the morphological differences before and after optimization. All fiber estimates approximate the contained fibers well visually, but there are apparent differences between them. The cuPARE detection is quite fragmented, especially in the areas around the volume border. Similar problems appear in the result of the FCP-TM, but there a fewer fragments. This is likely due to the difference in clustering of the fiber objects. The best single fiber detection is achieved with InsegtFibre, but here some fibers are completely missing in the lower right corner. This is due to the fact that InsegtFibre goes through the volume on a slice-by-slice basis and very sheer cuts through fibers lead to a worse detection of the fiber location. The cuPARE method and FCP-TM consider the whole 3D volume for analysis, which mitigates that problem, but brings additional clustering complexity.

The InsegtFibre result was chosen as the de-facto ground truth, because the fibers were detected on the full resolution reconstruction and should therefore be more accurate. The direct comparison was done on the five fibers with the highest overlap between all models from the entire fiber population. This means that all methods have almost identical starting conditions before the optimization. The fiber models obtained with FCP-TM and cuPARE were optimized based on the simulated projection data as described above. With the added complexity of real data, the number of projections was increased to 50 and the convergence threshold for the Nelder-Mead optimizer was set to 5e-9, two orders of magnitude lower than for the simulation experiments. The maximum number of iterations was increased to 2000.

Figure 4.26 shows a render of the per-vertex Euclidean distances between the surfaces of the mesh models of cuPARE and FCP-TM relative to the result of InsegtFibre. The initial model (seen in Subfigures (b) and (d)) shows good agreement in the center of the fiber in both cases, but at the ends the detected fiber deviates with increasing magnitude from the center towards the end. In particular, the first fiber from the bottom shows great deviation in Subfigure (b). The optimization procedure manages to even out the deviations (Subfigure (c)), which leads to a better agreement with the reference overall. At the

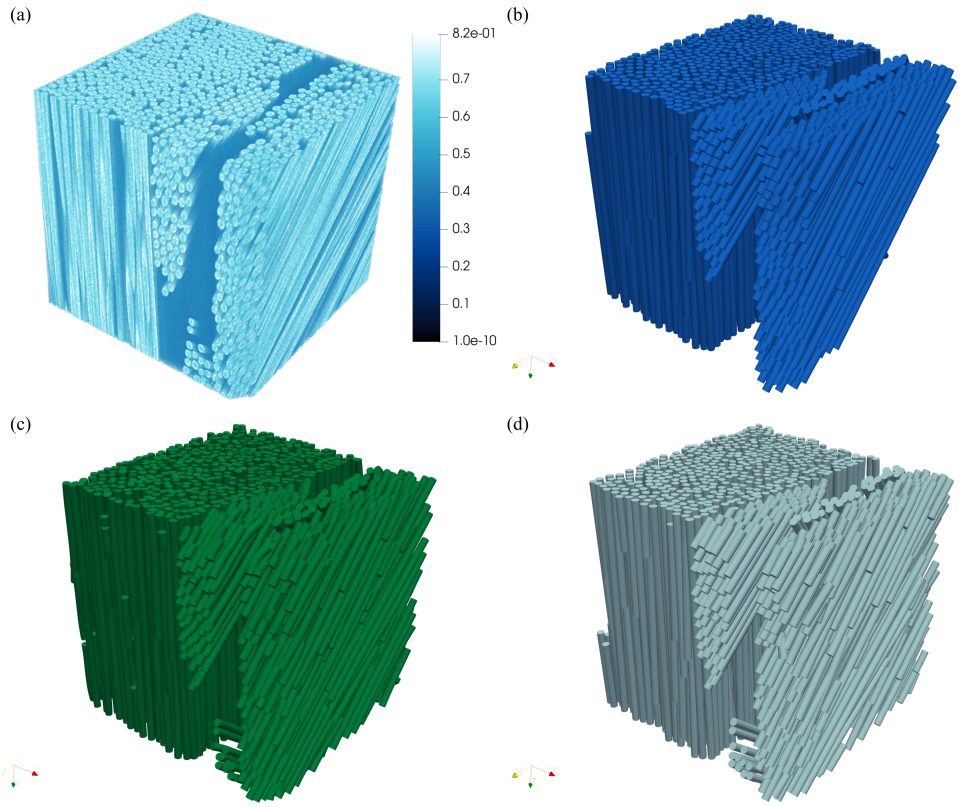


FIGURE 4.25: Comparison of the traces of dataset XCT_H. (a) shows the full sized reconstruction as in Figure 4.23a, (b) shows a rendering of the fiber CAD models acquired from the traces from InsegtFibre, (c) the fiber models obtained with cuPARE and (d) the fiber models obtained with FCP-TM. Figure reproduced from [106].

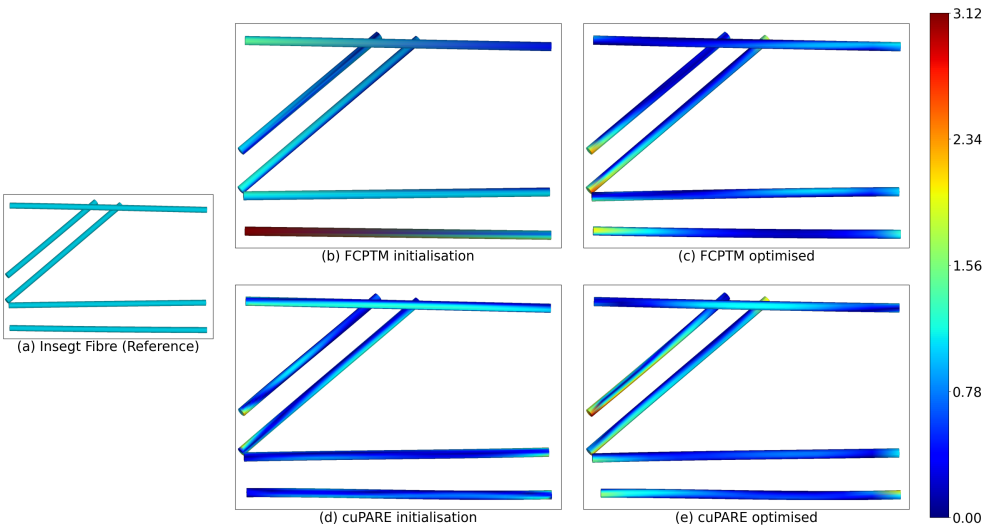


FIGURE 4.26: Visualization of the per-vertex Euclidean distance between the reference fiber models, the models from the fiber detection by FCP-TM and cuPARE and the optimized models of both of those methods. Note that the orientation of the fibers is changed compared to the previous figures to allow larger images of the meshes. Figure reproduced from [106].

same time it also causes small deviations in the center region of the fibers (see Subfigure (e)). Overall the optimization increases the agreement with the reference model, though the diagonal fibers are not moving or even get pushed away from the reference. The average distance from the reference mesh is $\leq$ 1 voxel, but can reach values up to 3.12 voxels, especially at the fiber ends.

The detection results from the dataset with the simulated curvature are shown in Figure 4.27. With cuPARE 1291 fibers were detected in the volume, InsegtFibre had 852 fibers. The FCP-TM method was not applied to this dataset as it was designed for straight fibers. In the volume rendering Figure 4.27 a lot more small fiber fragments can be seen compared to the unaltered reconstruction. Also in this case it happens mostly at the volume borders. The effect is reinforced by the fact that the strong curvature splits fibers into smaller segments that intersect at a sheer angle with the volume border. The detection in the center of the volume has a very good agreement with the reconstruction and the overall curvature of the twisted fibers is comparable to the source dataset.

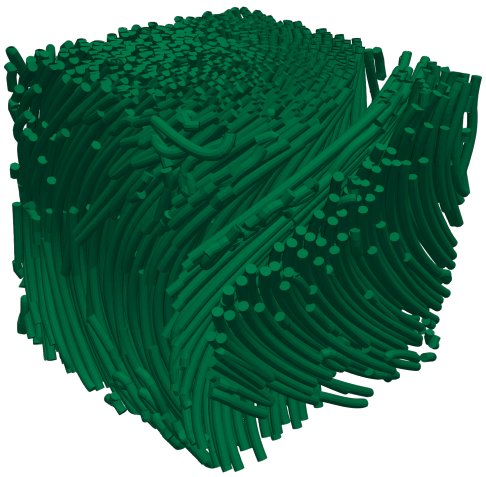


FIGURE 4.27: Rendering of the fibers in the real dataset shown in Figure 4.23b detected by cuPARE. Figure reproduced from [106].

The five fibers with the smallest *overlap dissimilarity* between InsegtFibre and cuPARE were optimized and then compared as in the previous experiment by computing the per-vertex Euclidean distances between the fiber meshes. This is shown in Figure 4.28. The fiber models are on average further away from their reference and improve at the fiber ends after optimization. This indicates that the estimates are slightly further off the reference than for the straight fibers. While the agreement between the reference and the estimates improves in some areas, the overall optimization is less effective compared to the straight fibers in the same dataset. This most likely has to do with the additional complexity of the curvature, as the fibers follow an S-curve, which requires more iterations to be changed in all degrees of freedom.

4.7 Discussion

The cuPARE algorithm is able to detect curved as well as straight fibers in simulated and realistic data and create a mathematical model of the individual fibers in those datasets. It was shown that it is possible to estimate the shape of those individual fibers based on very few projections and retrieve model parameters with good agreement to the ground truth. For real data it

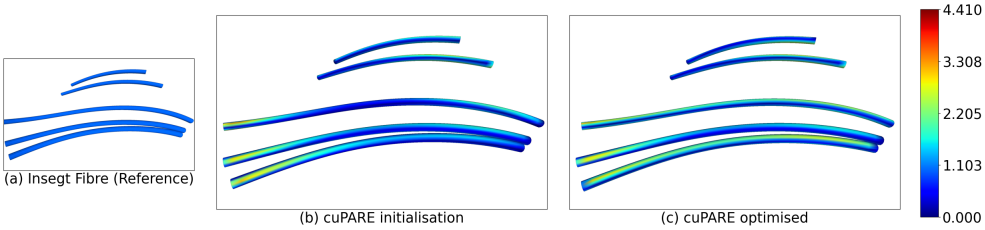


FIGURE 4.28: Visualization of the per-vertex Euclidean distance between the reference fiber models, the models from the fiber detection by cuPARE and their optimized counterparts from the curved version of the real dataset shown in Figure 4.23b. Figure reproduced from [106].

could be shown that the morphology of the fibers can be adjusted well using the presented approach, compared to a reference method used on a higher resolution reconstruction.

In terms of computational efficiency, improvements could be made by directly computing a projection image from the coefficients of the center line. However, this would require a tailor-made projector for such objects, which is currently not available to the author's knowledge. Such a change would also improve the stability of the optimization by introducing less in-directions from the model to the projections, which all can propagate errors through the individual steps. The current way of generating the fiber model using the VTK tube filter [126] can produce jumps in the objective function for small variations, which decreases numerical stability of any optimization procedure and forces the user to iterate for longer.

Chapter 5

General Conclusions and Future Perspectives

In this thesis, a new framework for detection and modeling of individual fibers in GFRP was introduced. Starting from a simplified, straight fiber model (chapter 3), the framework was extended to curved fibers and applied to real tomographic scans of GFRP (chapter 4). It was shown that it is able to reliably detect the individual fibers in a reconstructed volume, given some initial knowledge about the fiber attenuation and the radius. The fiber models are close to the original reconstruction data and are matched to the raw scan data via numerical optimization based on the projections. The procedure works even in the presence of noise and with very few projections, which reduces data and computational requirements. The framework was presented using the example of GFRP, but is general in nature. Further experiments could be conducted in the applicability to related problems. Without adjusting the model or processing pipeline, it should be possible to apply cuPARE to fiber bundles or concrete reinforced with steel bars, given that the reconstruction quality is sufficient. The framework should also naturally extend to having the fiber radius as an estimated parameter.

The introduction of an additional data "cleaning" step before the fiber detection step could further improve performance on real data. Artifacts like beam-hardening, streaks and partial volume effects are currently not addressed. In the work of Pelt et al. [131], for example Multi-Scale Dense convolutional Neural Networks (MSDNN) are used to improve reconstructions made with limited data. A number of experiments regarding the improvement of the SNR using MSDNN was already presented at the ICTMS conference in 2019 [132].

The general nature of the framework also allows replacement of different steps and models, possibly enabling analysis of more complex composites or

classification of defects like pores, foreign inclusions, fibers made from different materials or woven fiber bundles [133]. To that end, the model would have to be extended to be able to represent the new types of objects and the detection and clustering algorithms would need to be adjusted to differentiate between the objects reliably. A recently introduced neural network for segmentation called Segment Anything Model (SAM) [134] could be used as a starting point for the detection in such a case, given further research into the topic.
Due to the large amount of data, the method unfortunately scales poorly to large amounts of fibers. Here, the introduction of a numerical optimization based on derivatives, if possible, could prove helpful. That, as well as an effort to optimize all fiber parameters at once could significantly reduce the required amount of iterations as well as improve numerical stability. This, of course, would require a serious research effort, as the parameter space would grow extremely large. Another way to improve the stability of the optimization, would be to change the optimization model entirely by computing the analytic derivatives per vertex of the fiber mesh and then moving the vertices accordingly. This would enable a smooth transition between two fibers that are close in parameter space, but would require coming up with a way of converting a mesh back to a fiber model, which currently is not possible in the presented framework. Koo et al. [135] lay the groundwork for such an approach and Renders et al. [136] already apply a similar method to 4D CT images of foam.

The work presented in this thesis, like most scientific work, shows several ways forward in this direction of the field of NDT. The framework provided by cuPARE provides ample opportunity to develop better models, more efficient algorithms and make the technique applicable in an industry setting.

Appendix A

Scientific Contributions

Journal articles

1. **Tim Elberfeld**, Bernhard Fröhler, Christoph Heinzl, Jan Sijbers, Jan De Beenhouwer, *"cuPARE: Parametric Reconstruction of Curved Fibers from Glass fiber-reinforced Composites"*, Nondestructive Testing and Evaluation, pp. 1–20, 2022. https://doi.org/10.1080/10589759.2022.2155647

   Personal Contribution:
   Conception of methodology and experiments, implementation of the source code, data analysis, preparation, editing and reviewing of the manuscript.

2. Bernhard Fröhler, **Tim Elberfeld**, Torsten Möller, Hans-Christian Hege,, Julia Maurer, Christoph Heinzl, *"Sensitive vPSA – Exploring Sensitivity in Visual Parameter Space"*, preprint, 2022. https://doi.org/10.48550/arXiv.2204.01823

   Personal Contribution:
   Testing of the software tool, generation of test data, data analysis, editing and review of the manuscript.

3. Björn De Samber, Jens Renders, **Tim Elberfeld**, Yves Maris, Jonathan Sanctorum, Nathanaël Six, Zhihua Liang, Jan De Beenhouwer, Jan Sijbers, *"FleXCT: a Flexible X-ray CT scanner with 10 degrees of freedom"*, Optics Express, vol.29, pp. 3438–3457, 2021. https://doi.org/10.1364/OE.409982

   Personal Contribution:
   Writing of software, acquisition of data, editing and review of the manuscript.

4. Bernhard Fröhler, **Tim Elberfeld**, Torsten Möller, Hans-Christian Hege, Jan De Beenhouwer, Jan Sijbers, Johann Kastner, Jan Sijbers, *"Analysis and comparison of algorithms for the tomographic reconstruction of curved fibres"*, Nondestructive Testing and Evaluation , vol.35, Issue 3: iCT2020 Special Issue: NTE, 2020. https://doi.org/10.1080/10589759.2020.1774583

   Personal Contribution:
   Testing of the software, data analysis, editing and review of the manuscript.

5. Bernhard Fröhler, **Tim Elberfeld**, Torsten Möller, Hans-Christian Hege, Johannes Weissenböck, Jan De Beenhouwer, Jan Sijbers, Johann Kastner, Christoph Heinzl, *"A Visual Tool for the Analysis of Algorithms for Tomographic Fiber Reconstruction in Materials Science"*, Computer Graphics Forum , vol.38, pp.273–283, 2019. https://doi.org/10.1111/cgf.13688

   Personal Contribution:
   Testing of the software tool, writing of software, generation of test data, data analysis, editing and review of the manuscript.

6. **Tim Elberfeld**, Jan De Beenhouwer, Arnold Jan den Dekker, Christoph Heinzl, Jan Sijbers, *"Parametric Reconstruction of Glass Fiber-reinforced Polymer Composites from X-ray Projection Data – A Simulation Study"*, Journal of Nondestructive Evaluation, vol.37, no.62, 2018. https://doi.org/10.1007/s10921-018-0514-0

   Personal Contribution:
   Conception of methodology and experiments, implementation of the source code, data analysis, preparation, editing and reviewing of the manuscript.

Conference proceedings

1. Pavel Paramonov, Jens Renders, **Tim Elberfeld**, Jan De Beenhouwer, Jan Sijbers, *"Efficient X-ray projection of triangular meshes based on ray tracing and rasterization"*, SPIE Optical Engineering: Developments in X-Ray Tomography XIV, 2022

   Personal Contribution:

Methodology, implementation of source code, data analysis, editing and reviewing of the manuscript.

2. Bernhard Fröhler, **Tim Elberfeld**, Torsten Möller, Hans-Christian Hege, Jan De Beenhouwer, Jan Sijbers, Johann Kastner, Christoph Heinzl, *"Analysis and Comparison of Algorithms for the Tomographic Reconstruction of Curved Fibers"*, 10th Conference on Industrial Computed Tomography (ICT 2020), 2020

   Personal Contribution:
   Testing of the software, data analysis, editing and review of the manuscript.

3. **Tim Elberfeld**, Jan De Beenhouwer, Jan Sijbers, *"Fiber assignment by continuous tracking for parametric fiber reinforced polymer reconstruction"*, 15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine (Fully3D),vol. 11072, 2019, http://dx.doi.org/10.1117/12.2534836

   Personal Contribution:
   Conception of methodology and experiments, implementation of the source code, data analysis, preparation, editing and reviewing of the manuscript.

4. **Tim Elberfeld**, Shabab Bazrafkan, Jan De Beenhouwer, Jan Sijbers, *"Mixed-Scale Dense Convolutional Neural Network based Improvement of Glass Fiber-reinforced Composite CT Images"*, 4th International Conference on Tomography of Materials & Structures, 2019

   Personal Contribution:
   Conception of methodology and experiments, training data generation, data analysis, writing, editing and reviewing of the manuscript

5. **Tim Elberfeld**, Jan De Beenhouwer, Arnold Jan den Dekker, Christoph Heinzl, Jan Sijbers, *"Parametric Reconstruction of Advanced Glass Fiber-reinforced Polymer Composites from X-ray Images"*, 8th Conference on Industrial Computed Tomography. Wels, Austria, 2018

   Personal Contribution:
   Conception of methodology and experiments, implementation of the source code, data analysis, preparation, editing and reviewing of the manuscript.

Appendix B

Abbreviations

μCT	micro-computed tomography
4D CT	4-dimensional (i.e. time-resolved) computed tomography
AABB	axis-aligned bounding box
ASTRA	All Scale Tomographic Reconstruction Antwerp /Amsterdam
BB	Barzilai-Borwein
CSR	Cosine Series Representation
CT	computed tomography
cuPARE	curved Parametric Reconstruction
FACT	Fiber Assignment by Continuous Tracking
FBP	filtered backprojection
FCP-TM	Fiber Characterization Pipeline "Template Matching"
FDK	Feldkamp, Davis, and Kress
FOV	field of view
GFRP	glass fiber-reinforced polymer
GPU	graphical processing unit
ITK	Insight Toolkit
MRI	Magnetic Resonance Imaging

MSDNN	Multi-Scale Dense convolutional Neural Networks
NCC	normalized cross-correlation
NDT	non-destructive testing
NMR	nuclear magnetic resonance
PARE	Parametric Reconstruction
PCBs	printed circuit boards
PSNR	peak signal-to-noise ratio
ROI	region of interest
RSA	random sequential adsorption
RVE	representative volume element
SAM	Segment Anything Model
SDD	source-detector distance
SIRT	simulataneous iterative reconstruction technique
SNR	signal-to-noise ratio
SOD	source-object distance
VMF	Van Mises - Fisher distribution
VTK	Visualization Toolkit
XCT	X-ray computed tomography

Bibliography

[1] Blender Online Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[2] F. T. Wallenberger, J. C. Watson, and H. Li, "Glass fibers," in *ASM Handbook: Composites*. ASM International, 2001, vol. 21, pp. 27–34. [Online]. Available: https://doi.org/10.31399/asm.hb.v21.a0003353

[3] M. J. Fagan and M. Postema, *Introduction to stress and strain analysis*. Kingston upon Hull: University of Hull, Jun. 2007. [Online]. Available: https://doi.org/10.5281/zenodo.7503946

[4] L. E. Samuels, *Light Microscopy of Carbon Steels*. ASM International.

[5] S. Carmignato, W. Dewulf, and R. Leach, Eds., *Industrial X-Ray Computed Tomography*, 1st ed. Springer International Publishing, 2018.

[6] T. M. Buzug, *Computed Tomography - From Photon Statistics to Modern Cone-Beam CT*. Berlin, Heidelberg: Springer-Verlag, 2008.

[7] T. Elberfeld, "Implementation of phase unwrapping algorithms for coherentX-ray diffraction imaging," Master's thesis, University of Applied Sciences Darmstadt, h_da, Darmstadt, Germany, 2016, unpublished.

[8] B. D. Samber, J. Renders, T. Elberfeld, Y. Maris, J. Sanctorum, N. Six, Z. Liang, J. De Beenhouwer, and J. Sijbers, "FleXCT: a Flexible X-ray CT scanner with 10 degrees of freedom," *Optics Express*, vol. 29, pp. 3438–3457, 2021. [Online]. Available: https://doi.org/10.1364/OE.409982

[9] M. MacNeil and D. McMillan, "An introduction to 3d microscopy techniques," *Microscopy Today*, vol. 16, no. 1, pp. 20–23, 2008.

[10] G. Collingridge, "The brain slice preparation: a tribute to the pioneer Henry McIlwain," *Journal of Neuroscience Methods*, vol. 59, pp. 5–9, 1995. [Online]. Available: https://doi.org/10.1016/0165-0270(94)00187-l

[11] D. Peterson, "Stereology," in *Encyclopedia of Movement Disorders*, K. Kompoliti and L. V. Metman, Eds. Oxford: Academic Press, 2010, pp. 168–170. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123741059002872

[12] J. Ohser and K. Schladitz, *3D Images of Materials Structures*. Weinheim: WILEY-VCH Verlag GmbH & Co. KGaA, 2009. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527628308

[13] M. Ivanov, A. Ulanov, and N. Cherkasov, "Visual control of weld defects using computer vision system on fanuc robot," in *Proceedings - 2022 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2022*, 2022, pp. 859 — 863. [Online]. Available: https://dx.doi.org/10.1109/ICIEAM54945.2022.9787245

[14] P. M. Griffin, J. Villalobos, J. W. Foster, and S. L. Messimer, "Automated visual inspection of bare printed circuit boards," *Computers & Industrial Engineering*, vol. 18, no. 4, pp. 505–509, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/036083529090008A

[15] M. Wortberg, "Blister packing control in practice. Use of color image processing systems in the pharmaceutical industry; [Blister-Füllgutkontrolle in der Praxis. Einsatzmöglichkeiten von Farb-Bildverarbeitungssystemen in der pharmazeutischen Industrie]," *Pharmazeutische Industrie*, vol. 56, no. 1, pp. 55 — 58, 1994.

[16] J. R. Lakowicz, *Principles of fluorescence spectroscopy*. New York, NY: Springer, 2006. [Online]. Available: https://dx.doi.org/10.1007/978-0-387-46312-4

[17] J. R. Edwards, H. Ruparel, and J. Jingyue, "Mass-spectrometry dna sequencing," *Mutation Research*, vol. 1-2, pp. 3–12, 2005. [Online]. Available: https://doi.org/10.1016/j.mrfmmm.2004.07.021

[18] C. Zhang, C. Guo, F. Liu, W. Kong, Y. He, and B. Lou, "Hyperspectral imaging analysis for ripeness evaluation of strawberry with support vector machine," *Journal of Food Engineering*, vol. 179, pp. 11–18, 2016.

[Online]. Available: https://www.sciencedirect.com/science/article/pii/S0260877416300024

[19] M. A. Mohd Shahrimie, S. Mertens, L. Verbraeken, S. Dhondt, D. Inze, B. Koirala, and P. Scheunders, "Non-destructive analysis of plant physiological traits using hyperspectral imaging: A case study on drought stress," *Computers and Electronics in Agriculture*, vol. 195, p. 106806, 2022. [Online]. Available: https://dx.doi.org/10.1016/j.compag.2022.106806

[20] L. Vincze, B. Vekemans, F. Brenker, G. Falkenberg, K. Rickers, A. Somogyi, M. Kersten, and F. Adams, "Three-dimensional trace element analysis by confocal x-ray microfluorescence imaging," *Analytical Chemistry*, vol. 76, no. 22, pp. 6786–6791, 2004. [Online]. Available: https://doi.org/10.1021/ac049274l

[21] B. Goris, J. De Beenhouwer, A. De Backer, D. Zanaga, K. J. Batenburg, A. Sánchez-Iglesias, L. M. Liz-Marzán, S. Van Aert, S. Bals, J. Sijbers, and G. Van Tendeloo, "Measuring Lattice Strain in Three Dimensions through Electron Microscopy," *Nano Letters*, vol. 15, no. 10, pp. 6996–7001, 2015.

[22] W. Oliver and G. Pharr, "Measurement of hardness and elastic modulus by instrumented indentation: Advances in understanding and refinements to methodology," *Journal of Materials Research*, vol. 19, no. 1, pp. 3 — 20, 2004. [Online]. Available: https://dx.doi.org/10.1557/jmr.2004.19.1.3

[23] C. Stephen, S. R. Behara, B. Shivamurthy, R. Selvam, S. Kannan, and M. Abbadi, "Finite element study on the influence of fiber orientation on the high velocity impact behavior of fiber reinforced polymer composites," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 16, pp. 459 – 468, 2022. [Online]. Available: https://doi.org/10.1007/s12008-021-00808-7

[24] H. A. Stuart and G. Klages, *Kurzes Lehrbuch der Physik*. Springer Berlin Heidelberg, 2010. [Online]. Available: https://dx.doi.org/10.1007/978-3-540-89046-1

[25] J. Als-Nielsen and D. McMorrow, *Elements of modern x-ray physics*. New York, NY: Wiley, 2001. [Online]. Available: https://cds.cern.ch/record/479075

[26] B. D. Cullity, *Elements of X-Ray Diffraction*. Addison-Wesley Publishing Company Inc., 1956. [Online]. Available: https://archive.org/details/elementsofxraydi030864mbp

[27] D. Laguitton and W. Parrish, "Experimental spectral distribution versus Kramers' law for quantitative x-ray fluorescence by the fundamental parameters method," *X-Ray Spectrometry*, vol. 6, no. 4, pp. 201–203, 1977. [Online]. Available: https://dx.doi.org/10.1002/xrs.1300060409

[28] W. Duane and F. L. Hunt, "On x-ray wave-lengths," *Physical Review*, vol. 6, pp. 166–172, August 1915. [Online]. Available: https://dx.doi.org/10.1103/PhysRev.6.166

[29] H. G. J. Moseley, "The high-frequency spectra of the elements," *Philosophical Magazine Series 6*, vol. 26, no. 156, pp. 1024–1034, December 1913. [Online]. Available: https://dx.doi.org/10.1080/14786441308635052

[30] R. Brooks and G. Di Chiro, "Beam hardening in x-ray reconstructive tomography," *Physics in Medicine & Biology*, vol. 21, p. 390, 1976. [Online]. Available: https://doi.org/10.1088/0031-9155/21/3/004

[31] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. [Online]. Available: http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X

[32] G. Held, *Introduction to Light Emitting Diode Technology and Applications*. CRC Press, Taylor & Francis, 2009.

[33] W. Müller and H. Geiger, "Elektronenzählrohr zur Messung schwächster Aktivitäten," *Naturwissenschaften*, vol. 16, pp. 617–618, 1928. [Online]. Available: https://doi.org/10.1007/BF01494093

[34] J. Prince and J. Links, *Medical Imaging Signals and Systems*. Pearson, 2015.

[35] P. Toft, "The radon transform - theory and implementation," Ph.D. dissertation, Technical University of Denmark, 1996.

[36] P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. Patent 30696541962, 1962.

[37] L. A. Shepp and B. F. Logan, "The fourier reconstruction of a head section," *IEEE Transactions on Nuclear Science*, vol. 21, no. 3, pp. 21–43, 1974. [Online]. Available: https://dx.doi.org/10.1109/TNS.1974.6499235

[38] A. C. Kak and M. Slaney, *Principles of computerized tomographic imaging*. Philadelphia: Society for Industrial and Applied Mathematics, 2001. [Online]. Available: https://doi.org/10.1137/1.9780898719277

[39] L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone-beam algorithm," *Journal of the Optical Society of America A*, vol. 1, no. 6, pp. 612–619, Jun 1984. [Online]. Available: http://opg.optica.org/josaa/abstract.cfm?URI=josaa-1-6-612

[40] D. L. Parker, "Optimal short scan convolution reconstruction for fan beam ct," *Medical Physics*, vol. 9, no. 2, pp. 254–257, 1982. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.595078

[41] M. J. Lagerwerf, D. M. Pelt, W. J. Palenstijn, and K. J. Batenburg, "A computationally efficient reconstruction algorithm for circular cone-beam computed tomography using shallow neural networks," *Journal of Imaging*, vol. 6, no. 12, 2020. [Online]. Available: https://www.mdpi.com/2313-433X/6/12/135

[42] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer-Verlag, 2006.

[43] P. C. Hansen, J. Jørgensen, and W. R. B. Lionheart, *Computed Tomography: Algorithms, Insight, and Just Enough Theory*, P. C. Hansen, J. Jørgensen, and W. R. B. Lionheart, Eds. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2021. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611976670

[44] P. C. Hansen and J. S. Jørgensen, "Air tools II: algebraic iterative reconstruction methods, improved implementation," *Numerical Algorithms*, vol. 79, pp. 107 – 137, 2018. [Online]. Available: https://doi.org/10.1007/s11075-017-0430-x

[45] N. Six, J. Renders, J. Sijbers, and J. D. Beenhouwer, "Newton-Krylov methods for polychromatic X-ray ct," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3045–3049.

[46] J. Barzilai and J. M. Borwein, "Two-Point Step Size Gradient Methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 01 1988. [Online]. Available: https://doi.org/10.1093/imanum/8.1.141

[47] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, K. J. Batenburg, and J. Sijbers, "Fast and flexible X-ray tomography using the ASTRA toolbox," *Optics Express*, vol. 24, no. 22, pp. 25 129–25 147, Oct 2016. [Online]. Available: https://doi.org/10.1364/OE.24.025129

[48] P. Silverman, C. Cooper, D. Weltman, and R. Zeman, "Helical ct: practical considerations and potential pitfalls," *Radiographics*, vol. 15, pp. 25–36, 1995. [Online]. Available: https://doi.org/10.1148/radiographics.15.1.7899611

[49] V. Van Nieuwenhove, J. De Beenhouwer, J. Vlassenbroeck, M. Brennan, and J. Sijbers, "MoVIT: a tomographic reconstruction framework for 4d-ct," *Optics Express*, vol. 25, no. 16, pp. 19 236–19 250, Aug 2017. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-25-16-19236

[50] S. Gondrom, J. Zhou, M. Maisl, H. Reiter, M. Kröning, and W. Arnold, "X-ray computed laminography: an approach of computed tomography for applications with limited access," *Nuclear Engineering and Design*, vol. 190, no. 1, pp. 141–147, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029549398003197

[51] G. R. Myers, A. M. Kingston, T. K. Varslot, M. L. Turner, and A. P. Sheppard, "Dynamic tomography with a priori information," *Applied Optics*, vol. 50, no. 20, pp. 3685–3690, 2011. [Online]. Available: https://opg.optica.org/ao/abstract.cfm?URI=ao-50-20-3685

[52] T. Baranowski, D. Dobrovolskij, K. Dremel, A. Hölzing, G. Lohfink, K. Schladitz, and S. Zabler, "Local fiber orientation from X-ray region-of-interest computed tomography of large fiber reinforced composite components," *Composites Science and Technology*, vol. 183, p. 107786, 2019. [Online]. Available: https://doi.org/10.1016/j.compscitech.2019.107786

[53] R. D. Jean Lemaitre, *Engineering Damage Mechanics - Ductile, Creep, Fatigue and Brittle Failures*, 1st ed. Springer Berlin, Heidelberg, 2005. [Online]. Available: https://link.springer.com/book/10.1007/b138882

[54] C. Savandaiah, J. Maurer, M. Gall, A. Haider, G. Steinbichler, and J. Sapkota, "Impact of processing conditions and sizing on the thermomechanical and morphological properties of polypropylene/carbon fiber composites fabricated by material extrusion additive manufacturing," *Journal of Applied Polymer Science*, vol. 138, no. 16, p. 50243, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/app.50243

[55] E. Fitzer, R. Kleinholz, H. Tiesler, M. H. Stacey, R. De Bruyne, I. Lefever, A. Foley, W. Frohs, T. Hauke, M. Heine, H. Jäger, and S. Sitter, *Fibers, 5. Synthetic Inorganic*. John Wiley & Sons, Ltd, 2008. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/14356007.a11_001.pub2

[56] G. Erhard, "Designing for production," in *Designing with Plastics*, G. Erhard, Ed. Hanser, 2006, pp. 239–309. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9783446225909500088

[57] U. Strautins, "On stability of a concentrated fiber suspension flow," in *Progress in Industrial Mathematics at ECMI 2012*, M. Fontes, M. Günther, and N. Marheineke, Eds. Cham: Springer International Publishing, 2014, pp. 127–133. [Online]. Available: https://doi.org/10.1007/978-3-319-05365-3_17

[58] H. Altendorf, "3d morphological analysis and modeling of random fiber networks," Ph.D. dissertation, École nationale supérieure des mines de Paris and Technische Universität Kaiserslautern – Fachbereich Mathematik, 2011.

[59] H. Altendorf and D. Jeulin, "3d directional mathematical morphology for analysis of fiber orientations," *Image Analysis & Stereology*, vol. 28, no. 3, pp. 143–153, 2011. [Online]. Available: https://www.ias-iss.org/ojs/IAS/article/view/857

[60] H. Andrä, M. Gurka, M. Kabel, S. Nissle, C. Redenbach, K. Schladitz, and O. Wirjadi, "Geometric and mechanical modeling of fiber-reinforced composites," in *Proceedings of the 2nd International Congress on 3D Materials Science*, D. Bernard, J.-Y. Buffière, T. Pollo-ck, H. F. Poulsen, A. Rollett, and M. Uchic, Eds. Cham: Springer International Publishing, 2016, pp. 35–40.

[61] O. Wirjadi, K. Schladitz, P. Easwaran, and J. Ohser, "Estimating fibre direction distributions of reinforced composites from tomographic images," *Image Analysis & Stereology*, vol. 35, no. 3, pp. 167–179, 2016. [Online]. Available: https://www.ias-iss.org/ojs/IAS/article/view/1489

[62] P. Pinter, B. Bertram, and K. A. Weidenmann, "A novel method for the determination of fibre length distributions from μct-data," *Proceedings of the 6th Conference on Industrial Computed Tomography*, 2016.

[63] P. Townsend, E. Larsson, T. Karlson, S. A. Hall, M. Lundman, P. Bergström, C. Hanson, N. Lorén, T. Gebäck, A. Särkkä, and M. Röding, "Stochastic modelling of 3d fiber structures imaged with x-ray microtomography," *Computational Materials Science*, vol. 194, p. 110433, 2021. [Online]. Available: https://doi.org/10.1016/j.commatsci.2021.110433

[64] K. Sandau and J. Ohser, "The chord length transform and the segmentation of crossing fibres," *Journal of microscopy*, vol. 226, pp. 43–53, 2007. [Online]. Available: https://dx.doi.org/10.1111/j.1365-2966.2007.01748.x

[65] T. K. Konopczynski, D. Rathore, J. Rathore, T. Kröger, L. Zheng, C. S. Garbe, S. Carmignato, and J. Hesser, "Fully convolutional deep network architectures for automatic short glass fiber semantic segmentation from CT scans," in *8th Conference on Industrial Computed Tomography (iCT)*, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1901.01211

[66] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. [Online]. Available: https://doi.org/10.1109/TSMC.1979.4310076

[67] W. Burger and M. J. Burge, *Principles of Digital Image Processing: Advanced Methods*. Springer, 2009.

[68] M. W. Czabaj, M. L. Riccio, and W. W. Whitacre, "Numerical reconstruction of graphite/epoxy composite microstructure based on sub-micron resolution X-ray computed tomography," *Composites Science and Technology*, vol. 105, pp. 174–182, 2014.

[69] F. Sket, A. Enfedaque, C. Alton, C. González, J. M. Molina-Aldareguia, and J. Llorca, "Automatic quantification of matrix cracking and fiber rotation by X-ray computed tomography in shear-deformed carbon fiber-reinforced laminates," *Composites Science and Technology*, vol. 90, pp. 129–138, 2014.

[70] J. Weissenböck, A. Amirkhanov, W. Li, A. Reh, A. Amirkhanov, E. Groller, J. Kastner, and C. Heinzl, "FiberScout: an interactive tool for exploring and analyzing fiber reinforced polymers," *2014 IEEE Pacific Visualization Symposium*, pp. 153–160, 2014.

[71] M. J. Emerson, K. M. Jespersen, A. B. Dahl, K. Conradsen, and L. P. Mikkelsen, "Individual fibre segmentation from 3d x-ray computed tomography for characterising the fibre orientation in unidirectional composite materials," *Composites Part A: Applied Science and Manufacturing*, vol. 97, pp. 82–92, 2017.

[72] X. Huang, D. Wen, Y. Zhao, Q. Wang, W. Zhou, and D. Deng, "Skeleton-based tracing of curved fibers from 3d x-ray microtomographic imaging," *Results in Physics*, vol. 6, pp. 170–177, 2016.

[73] H. Zauner, D. Salaberger, C. Heinzl, and J. Kastner, "3d image processing for single fibre characterization by means of XCT," *Proceedings of the 14th International Congress for Stereology and Image Analysis - Acta Stereologica*, vol. 1, no. 1, 2015.

[74] A. Rigort, D. Günther, R. Hegerl, D. Baum, B. Weber, S. Prohaska, O. Medalia, W. Baumeister, and H.-C. Hege, "Automated segmentation of electron tomograms for a quantitative description of actin filament networks," *Journal of Structural Biology*, vol. 177, no. 1, pp. 135–144, jan 2012.

[75] J. Ohser, *Angewandte Bildverarbeitung und Bildanalyse*. München: Hanser Verlag, 2018. [Online]. Available: https://www.hanser-elibrary.com/doi/abs/10.3139/9783446453081.fm

[76] J. P. Lewis, "Fast template matching," *Vision Interface*, vol. 95, pp. 120–123, 1995.

[77] T. Elberfeld, J. De Beenhouwer, A. den Dekker, C. Heinzl, and J. Sijbers, "Parametric reconstruction of glass fiber-reinforced polymer composites from x-ray projection data - a simulation study," *Journal*

of Nondestructive Evaluation, vol. 37, no. 3, 2018. [Online]. Available: https://doi.org/10.1007/s10921-018-0514-0

[78] E. T. Whittaker, "XVIII.—on the functions which are represented by the expansions of the interpolation-theory," *Proceedings of the Royal Society of Edinburgh*, vol. 35, p. 181–194, 1915. [Online]. Available: https://doi.org/10.1017/S0370164600017806

[79] W. Burger and M. J. Burge, *Principles of Digital Image Processing: Core Algorithms*. Springer, 2009.

[80] C. Dalitz, T. Schramke, and M. Jeltsch, "Iterative Hough Transform for Line Detection in 3D Point Clouds Algorithm," *Image Processing On Line*, vol. 7, pp. 184–196, 2017.

[81] M. Jeltsch, C. Dalitz, and R. Pohle-Fröhlich, "Hough parameter space regularisation for line detection in 3d," *Proceedings of the 11th International Conference on Computer Vision Theory and Applications*, vol. 4, pp. 345–352, 2016.

[82] K. Roberts, "A new representation for a line," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 635–640, 1988.

[83] A. Kaufman and E. Shimony, "3d scan-conversion algorithms for voxel-based graphics," *Proceedings of the ACM Workshop on Interactive 3D Graphics*, pp. 45–75, October 1986.

[84] W. Burger and M. J. Burge, *Principles of Digital Image Processing: Fundamental Techniques*. Springer, 2009.

[85] H. Prautzsch, W. Boehm, and M. Paluszny, *B-spline representation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 59–75. [Online]. Available: https://doi.org/10.1007/978-3-662-04919-8_5

[86] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. Cambridge University Press, 2007.

[87] D. Kühlke, *Optik: Grundlagen und Anwendungen; mit Tabellen, Beispielen und Aufgaben mit Lösungen*. Harri Deutsch Verlag, 2004.

[88] S. W. Smith, *The Scientist & Engineer's Guide to Digital Signal Processing*, 1st ed. California Technical Pub., 1997.

[89] V. Bleichrodt, J. De Beenhouwer, J. Sijbers, and K. J. Batenburg, "Aligning projection images from binary volumes," *Fundamenta Informaticae*, vol. 135, pp. 1–22, 2014.

[90] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Emerald Group Publishing Limited, 1982.

[91] J. Feder, "Random sequential adsorption," *Journal of Theoretical Biology*, vol. 87, no. 2, pp. 237–254, 1980. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022519380903586

[92] C. Redenbach and I. Vecchio, "Statistical analysis and stochastic modelling of fibre composites," *Composites Science and Technology*, vol. 71, no. 2, pp. 107–112, jan 2011.

[93] M. Schneider, "The sequential addition and migration method to generate representative volume elements for the homogenization of short fiber reinforced plastics," *Computational Mechanics*, vol. 59, no. 2, pp. 247–263, 2017.

[94] H. Altendorf and D. Jeulin, "Random-walk-based stochastic modeling of three-dimensional fiber systems," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 83, no. 4, p. 041804, 2011. [Online]. Available: https://dx.doi.org/10.1103/PhysRevE.83.041804

[95] R. Goldman, "Intersection of two lines in three-space," in *Graphics Gems*, A. S. Glassner, Ed. Academic Press, Boston, 1990.

[96] J. Ketchel and P. Larochelle, "Collision detection of cylindrical rigid bodies for motion planning," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 1530–1535.

[97] D. Eberly, *Geometric Tools Library - Documentation: Robust Computation of Distance Between Line Segments*, 2022. [Online]. Available: https://www.geometrictools.com/Documentation/DistanceLine3Line3.pdf

[98] N. I. Fisher, *Statistical Analysis of Spherical Data*. Cambridge University Press, 1987. [Online]. Available: https://doi.org/10.1017/CBO9780511623059

[99] M. J. Emerson, V. A. Dahl, K. Conradsen, L. P. Mikkelsen, and A. B. Dahl, "A multimodal data-set of a unidirectional glass fibre reinforced polymer composite," *Data in Brief*, vol. 18, pp. 1388–1393, 2018.

[100] A. Bhattacharya, C. Heinzl, A. Amirkhanov, J. Kastner, and R. Wenger, "MetaTracts - A method for robust extraction and visualization of carbon fiber bundles in fiber reinforced composites," in *2015 IEEE Pacific Visualization Symposium (Pacific Vis)*, 2015, pp. 191–198. [Online]. Available: https://doi.org/10.1109/PACIFICVIS.2015.7156377

[101] M. Emerson, A. Dahl, K. Conradsen, and V. Dahl, "Insegt fibre: A user-friendly software for individual fibre segmentation," in *22nd International Conference on Composite Materials (ICCM22)*. Engineers Australia, 2019, pp. 1161 – 1172. [Online]. Available: https://search.informit.org/doi/10.3316/informit.863066300059491

[102] N. Adluru, A. L. Alexander, M. K. Chung, J. E. Lainhart, M. Lazar, and J. E. Lee, "Cosine series representation of 3D curves and its application to white matter fiber bundles in diffusion tensor imaging," *Statistics and Its Interface*, vol. 3, no. 1, pp. 69–80, 2010.

[103] A. Lemkaddem, D. Skiöldebrand, A. Dal Palú, J.-P. Thiran, and A. Daducci, "Global tractography with embedded anatomical priors for quantitative connectivity analysis," *Frontiers in Neurology*, vol. 5, p. 232, 2014. [Online]. Available: https://doi.org/10.3389/fneur.2014.00232

[104] S. Zhao, F. Luan, and K. Bala, "Fitting procedural yarn models for realistic cloth rendering," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–11, 2016. [Online]. Available: https://doi.org/10.1145/2897824.2925932

[105] A. Bhattacharjee and H. Roy, "Assessment of tensile and damping behaviour of hybrid particle/woven fibre/polymer composites," *Composite Structures*, vol. 244, p. 112231, 2020. [Online]. Available: https://doi.org/10.1016/j.compstruct.2020.112231

[106] T. Elberfeld, B. Fröhler, C. Heinzl, J. Sijbers, and J. De Beenhouwer, "cuPARE: Parametric reconstruction of curved fibers from glass fiber-reinforced composites," *Nondestructive Testing and Evaluation*, vol. 0, no. 0, pp. 1 – 20, 2022. [Online]. Available: https://doi.org/10.1080/10589759.2022.2155647

[107] P. J. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi, "In vivo fiber tractography using dt-mri data," *Magnetic Resonance in Medicine*, vol. 44, no. 4, pp. 625–632, 2000. [Online]. Available: https://doi.org/10.1002/1522-2594(200010)44:4<625::AID-MRM17>3.0.CO;2-O

[108] J. F. Epperson, "On the Runge example," *The American Mathematical Monthly*, vol. 94, no. 4, pp. 329–341, 1987. [Online]. Available: https://doi.org/10.2307/2323093

[109] B. Fröhler, T. Elberfeld, T. Möller, H.-C. Hege, J. Weissenböck, J. De Beenhouwer, J. Sijbers, J. Kastner, and C. Heinzl, "A visual tool for comparative analysis of algorithms for fiber characterization," *Computer Graphics Forum*, vol. 38, no. 3, pp. 1–11, 2019. [Online]. Available: https://doi.org/10.1111/cgf.13688

[110] T. Elberfeld, B. Fröhler, M. J. Emerson, C. Heinzl, J. Kastner, J. De Beenhouwer, and J. Sijbers, "Efficient parametric curved glass fiber representations," in *20th World Conference on Non-Destructive Testing (WCNDT)*, 2020, accepted.

[111] C. P. Schwegmann, W. Kleynhans, B. P. Salmon, and L. Mdakane, "Ship detection in sentinel-1 imagery using the h-dome transformation," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 3711–3714. [Online]. Available: https://doi.org/10.1109/IGARSS.2015.7326629

[112] T. Lee, R. Kashyap, and C. Chu, "Building skeleton models via 3-d medial surface axis thinning algorithms," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 6, pp. 462–478, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S104996528471042X

[113] H. Homann, "Implementation of a 3d thinning algorithm," *http://hdl.handle.net/1926/1292*, 2007.

[114] M. McCormick, X. Liu, L. Ibanez, J. Jomier, and C. Marion, "ITK: enabling reproducible research and open science," *Frontiers in Neuroinformatics*, vol. 8, 2014. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fninf.2014.00013

[115] R. Bellman, *Dynamic Programming*, ser. Dover Books on Computer Science Series. Dover Publications, 2003. [Online]. Available: https://books.google.be/books?id=fyVtp3EMxasC

[116] I. I. Rabi, J. R. Zacharias, S. Millman, and P. Kusch, "A new method of measuring nuclear magnetic moment," *Phys. Rev.*, vol. 53, pp. 318–318, Feb 1938. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.53.318

[117] E. L. Hahn, "Spin echoes," *Physical Review*, vol. 80, pp. 580–594, Nov 1950. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.80.580

[118] E. O. Stejskal and J. E. Tanner, "Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient," *The Journal of Chemical Physics*, vol. 42, no. 1, pp. 288–292, 1965. [Online]. Available: https://doi.org/10.1063/1.1695690

[119] S. Mori, B. J. Crain, V. P. Chacko, and P. C. M. Van Zijl, "Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging," *Annals of Neurology*, vol. 45, no. 2, pp. 265–269, 1999.

[120] T. Elberfeld, J. De Beenhouwer, and J. Sijbers, "Fiber assignment by continuous tracking for parametric fiber reinforced polymer reconstruction," in *Proc. SPIE 11072, 15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, 2019. [Online]. Available: https://doi.org/10.1117/12.2534836

[121] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach, "Ridges for image analysis," *Journal of Mathematical Imaging and Vision*, vol. 4, pp. 353–373, 1994. [Online]. Available: https://doi.org/10.1007/BF01262402

[122] J.-D. Tournier, R. Smith, D. Raffelt, R. Tabbara, T. Dhollander, M. Pietsch, D. Christiaens, B. Jeurissen, C.-H. Yeh, and A. Connelly, "Mrtrix3: A fast, flexible and open software framework for medical image processing and visualisation," *NeuroImage*, vol. 202, p. 116137, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811919307281

[123] M. Teßmann, S. Mohr, S. Gayetskyy, U. Haßler, R. Hanke, and G. Greiner, "Automatic determination of fiber-length distribution in composite material using 3D CT data," *Eurasip Journal on*

Advances in Signal Processing, no. 545030, 2010. [Online]. Available: https://dx.doi.org/10.1155/2010/545030

[124] F. Gao and L. Han, "Implementing the Nelder-Mead simplex algorithm with adaptive parameters," *Computational Optimization and Applications*, vol. 51, pp. 259–277, 2012. [Online]. Available: https://doi.org/10.1007/s10589-010-9329-3

[125] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. [Online]. Available: https://doi.org/10.1109/TIP.2003.819861

[126] W. Schroeder, K. Martin, and B. Lorensen, "The visualization toolkit (4th ed.)," *Kitware*, 2006.

[127] Q. Zhou, "Pymesh," 2021, retrieved 2021-10-12, commit `384ba882b`. [Online]. Available: https://github.com/PyMesh/PyMesh

[128] Á. Marinovszki, J. De Beenhouwer, and J. Sijbers, "An efficient CAD projector for X-ray projection based 3D inspection with the ASTRA Toolbox," in *8th Conference on Industrial Computed Tomography, Wels, Austria*, 2018, pp. 1 – 6.

[129] S. Singer and S. Singer, "Efficient implementation of the Nelder–Mead search algorithm," *Applied Numerical Analysis & Computational Mathematics*, vol. 1, no. 2, pp. 524–534, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/anac.200410015

[130] B. Fröhler, T. Elberfeld, T. Möller, H.-C. Hege, J. De Beenhouwer, J. Sijbers, J. Kastner, and C. Heinzl, "Analysis and comparison of algorithms for the tomographic reconstruction of curved fibres," *Nondestructive Testing and Evaluation*, vol. 35, no. 3, pp. 328–341, 2020. [Online]. Available: https://doi.org/10.1080/10589759.2020.1774583

[131] D. M. Pelt, K. J. Batenburg, and J. A. Sethian, "Improving tomographic reconstruction from limited data using mixed-scale dense convolutional neural networks," *Journal of Imaging*, vol. 4, no. 11, 2018. [Online]. Available: https://dx.doi.org/10.3390/jimaging4110128

[132] T. Elberfeld, S. Bazrafkan, J. De Beenhouwer, and J. Sijbers, "Mixed-Scale Dense Convolutional Neural Network based Improvement of

Glass Fiber-reinforced Composite CT Images," in *4th International Conference on Tomography of Materials & Structures*, 2019. [Online]. Available: http://ictms.p.ann.currinda.com/days/2019-07-23/abstract/132

[133] Y. Sinchuk, P. Kibleur, J. Aelterman, M. Boone, and W. Paepegem, "Geometrical and deep learning approaches for instance segmentation of CFRP fiber bundles in textile composites," *Composite Structures*, vol. 277, p. 114626, 2021. [Online]. Available: https://doi.org/10.1016/j.compstruct.2021.114626

[134] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2304.02643

[135] J. Koo, A. B. Dahl, J. A. Bærentzen, Q. Chen, S. Bals, and V. A. Dahl, "Shape from projections via differentiable forward projector for computed tomography," *Ultramicroscopy*, vol. 224, p. 113239, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304399121000334

[136] J. Renders, J. De Beenhouwer, and J. Sijbers, "Mesh-based reconstruction of dynamic foam images using x-ray ct," in *2021 International Conference on 3D Vision (3DV)*, 2021, pp. 1312–1320. [Online]. Available: https://dx.doi.org/10.1109/3DV53792.2021.00138